

MODULAR EMBEDDED SYSTEM DESIGN / IMPLEMENTATION FOR
MECHATRONICS EDUCATION AND RESEARCH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ALİ ÖZGÜ NURSAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

SEPTEMBER 2007

Approval of the thesis:

**MODULAR EMBEDDED SYSTEM DESIGN / IMPLEMENTATION FOR
MECHATRONICS EDUCATION AND RESEARCH**

submitted by **ALİ ÖZGÜ NURSAL** in partial fulfillment of the requirements
for the degree of **Master of Science in Mechanical Engineering, Middle
East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Kemal İder _____
Head of Department, **Mechanical Engineering Dept., METU**

Asst. Prof. Dr. Buğra Koku _____
Supervisor, **Mechanical Engineering Dept., METU**

Asst. Prof. Dr. Melik Dölen _____
Co-Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Reşit Soylu _____
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Buğra Koku _____
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Melik Dölen _____
Mechanical Engineering Dept., METU

Assoc. Prof. Dr. Veysel Gazi _____
Electrical and Electronics Engineering Dept., TOBB ETÜ

Asst. Prof. Dr. Yiğit Yazıcıoğlu _____
Mechanical Engineering Dept., METU

Date: 06 / 09 / 2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Ali Özgü Nursal

Signature :

ABSTRACT

MODULAR EMBEDDED SYSTEM DESIGN / IMPLEMENTATION FOR MECHATRONICS EDUCATION AND RESEARCH

Nursal, Ali Özgü

M.Sc., Department of Mechanical Engineering

Supervisor: Asst. Prof. Dr. Buğra Koku

Co-Supervisor: Asst. Prof. Dr. Melik Dölen

September 2007, 103 pages

In this thesis a modular embedded system for Mechatronics education and research is designed and implemented. Four types of control boards are manufactured and related software is developed at board and PC level.

A star like topology is used for boards architecture. One bridge board is responsible for handling communication between the PC and all the other boards that are connected independently to that bridge board. For PC communication Universal Serial Bus (USB), for inter peripheral communication serial peripheral interface (SPI) is used. By the use of USB and SPI, the system becomes scalable and hot pluggable.

All these four boards can be used in stand alone and/or together in different applications. The system covers; general analog digital input output, several types of motor control and simple graphical display applications.

The firmware and software is developed in a modular, flexible and transparent way allowing users all level accessibility. This way, the application fields and the content of the courses that will be using this system become very versatile.

A simulation of hard disk reading experiment is designed and implemented to show the integral capabilities of the system. In this experiment all the boards are utilized and used together.

As a result, a system is designed for Mechatronics experiments and research. A modular approach is adopted which is composed of units that can also work in a stand-alone manner. A software library developed for the PC platform facilitates high-level usage of these cards whereas it is possible to implement custom codes on these boards. To sum up, an affordable, versatile, scalable and transparent system is developed which is specifically tailored for Mechatronics applications.

Keywords: Mechatronic education, Mechatronic experiment, USB, modular embedded system

ÖZ

MEKATRONİK EĞİTİMİ VE ARAŞTIRMALARI İÇİN MODÜLER TÜMLEŞİK SİSTEM TASARIMI / UYGULANMASI

Nursal, Ali Özgü

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Y. Doç. Dr. Buğra Koku

Ortak Tez Yöneticisi: Y. Doç. Dr. Melik Dölen

Eylül 2007, 103 sayfa

Bu tezde mekatronik eğitiminde ve araştırmalarında kullanılmak üzere modüler tümleşik bir sistem tasarlanmış ve uygulanmıştır. Dört çeşit kontrol kartı üretilip, kart ve bilgisayar tarafında ilgili yazılımlar geliştirilmiştir.

Kartların mimarisinde yıldız tipi bir topoloji kullanılmıştır. Bir köprü kartı tüm sistemin bilgisayar haberleşmesinden sorumlu olup diğer kartlar bu köprü kartına bağımsız bir şekilde bağlanmaktadır. Bilgisayar haberleşmesi için Evrensel Seri Veriyolu (ESV), kartlar arası haberleşme için ise seri çevresel arayüzleme (SÇA) kullanılmıştır. ESV ve SÇA kullanımı sayesinde sistem ölçeklenebilir ve modüler olmuştur.

Bu dört kart kendi başlarına ve/veya birbirlerini tamamlayıcı olacak şekilde kullanılabilir. Sistem, genel analog dijital, çeşitli motor control ve basit grafik gösterme uygulamalarını kapsamaktadır.

Kullanıcıların yazılımsal olarak her seviyeye erişebilmesi için kart içi yazılım ve bilgisayar yazılımları modüler, esnek ve şeffaf bir şekilde geliştirilmiştir. Bu sayede

uygulamalarda ve bu kartı kullanacak derslerin içeriklerinde daha fazla çeşitlilik sağlanılabilir.

Sistemin tümleşik yeteneğini göstermek için sabit bellek okuma benzetimi yapan bir deney seti tasarlanıp uygulanmıştır. Bu deneyde tüm kartlar birlikte kullanılmıştır.

Sonuç olarak, mekatronik deneyleri ve araştırmaları için bir sistem tasarlanmıştır. Kendi başlarına da çalışabilen modüler bir sistem yaklaşımı uygulanmıştır. Yüksek seviyeli kart kontrolü için bilgisayarda yazılım kütüphanesi geliştirilmiştir. Ayrıca kartların üstündeki yazılımların değiştirilmesi de mümkündür. Toplamak gerekirse; özellikle mekatronik uygulamalarında kullanılmak için elde edilebilir, çeşitli, ölçeklenebilir ve şeffaf olan bir sistem tasarlanmıştır.

Anahatar Kelimeler: Mekatronik eğitimi, Mekatronik deneyi, ESV, modüler tümleşik sistem

To My Family

ACKNOWLEDGMENTS

I would like to thank for my supervisors Buğra Koku and Melik Dölen. This study is product of their supporting.

I am also grateful to my friends from department.

And I would like to express my sincere thanks to my family for everything.

TABLE OF CONTENTS

ABSTRACT.....	iv
ÖZ.....	vi
DEDICATION.....	viii
ACKNOWLEDGMENTS	ix
TABLE OF CONTENTS	x
LIST OF TABLES.....	xiv
LIST OF FIGURES.....	xv
CHAPTER	
1 INTRODUCTION.....	1
1.1 Mechatronic Related Courses Overview.....	2
1.1.1 ME 220 - Introduction to Mechatronics (METU).....	2
1.1.2 ME 407 - Mechanical Engineering Design (METU).....	3
1.1.3 ME 461 - Mechatronic Components and Instrument (METU).....	4
1.1.4 ME 534 Computer Control of Machines.....	4
1.1.5 ME307 - Mechatronics and Measurement Systems (Colorado State University).....	5
1.1.6 ME4447 - Microprocessor Control of Manufacturing Systems / ME6405 - Introduction to Mechatronics (Georgia Institute of Technology).....	6
1.1.7 ENGG3400/ENGG3490 - Introduction to Mechatronics Systems Design (University of Guelph).....	7
1.1.8 151-0641-00 - Introduction to Robotics & Mechatronics (Institute of Robotics and Intelligent Systems).....	9
1.1.9 2.737 - Mechatronics (Massachusetts Institute of Technology).....	9
1.1.10 ME333 - Introduction to Mechatronics (Northwestern University).....	10
1.1.11 ME210 – Introduction to Mechatronics	

	(Stanford University).....	11
1.2	Result of Course Survey.....	11
2	REVIEW OF THE STAE-OF-THE-ART.....	13
2.1	Boards Available at the Market.....	13
2.1.1	Quanser Products.....	13
2.1.2	Phidgets Products	15
2.1.3	Mikroelektronika Products	17
2.1.4	National Instruments Products	18
2.1.5	Result of Market Survey.....	20
2.2	Architecture.....	22
2.2.1	Computer Interfacing.....	23
2.2.1.1	Serial Port.....	23
2.2.1.2	Parallel Port.....	24
2.2.1.3	USB.....	24
2.2.1.4	Firewire.....	24
2.2.1.5	PCI Express.....	25
2.2.1.6	Comparison and Selection.....	25
2.2.2	Peripheral Boards.....	27
2.2.3	Peripheral Interfacing.....	27
2.2.3.1	I2C.....	28
2.2.3.2	SPI.....	28
2.2.3.3	Selection.....	29
2.3	Conclusion.....	30
3	USB BOARD.....	31
3.1	Design Stages and Hardware.....	31
3.2	Firmware.....	38
3.3	Conclusion.....	41
4	ANALOG INPUT DIGITAL INPUT OUTPUT BOARD.....	42
4.1	Hardware.....	42
4.2	Firmware.....	49
4.3	Conclusion.....	50

5	MOTOR BOARD.....	52
5.1	Hardware.....	52
5.2	Firmware.....	58
5.3	Conclusion.....	59
6	LCD BOARD.....	61
6.1	Hardware.....	61
6.2	Firmware.....	64
6.3	Conclusion.....	66
7	PC INTERFACING AND SOFTWARE DESIGN.....	68
7.1	DLL Methods.....	69
7.1.1	Debug Related Methods.....	71
7.1.2	Serial Port Related Methods.....	72
7.1.3	Updating Status of Boards Related Methods.....	72
7.1.3.1	USB Board Related Methods.....	72
7.1.3.2	AUDIO Board Related Methods.....	73
7.1.3.3	Motor Board Related Methods.....	73
7.1.3.4	LCD Board Related Methods.....	74
7.1.4	Getting Status of Board Related Methods.....	74
7.2	Sample GUI.....	75
7.3	Conclusion.....	78
8	CASE STUDY.....	79
8.1	Experiment Description.....	80
8.2	Experiment Procedure.....	81
8.3	Experiment Preparation and Results.....	82
9	CONCLUSION.....	86
9.1	Further Developments.....	87
	REFERENCES.....	89

APPENDICES

A	CD CONTENT.....	93
B	SIMULATION OF HARD DISK READING LAB SHEET.....	94

B.1	Objective.....	94
B.2	Procedure.....	95
B.2.1	Interfacing Reflective Optical Sensor and Switch.....	95
B.2.2	Counter.....	99
B.2.4	RC Servo Motor Control.....	99
B.2.3	Encoder Reading.....	100
B.2.5	DC Motor Driving.....	101

LIST OF TABLES

TABLE

1.1 Overview of courses and experiments.....	12
2.1 Overview of interfaces.....	29
3.1 PIC 18F4550 pin usages at USB Board.....	36
3.2 Ribbon cable connectors' pins' description.....	37
3.3 USB Board – PC communication protocol.....	40
4.1 PIC 16F877 pin usages at AIDIO Board.....	48
4.2 AIDIO Board – USB Board communication protocol.....	49
5.1 PIC 30F4011 pin usages at Motor Board.....	57
5.2 Motor Board – USB Board communication protocol.....	58
6.1 PIC 16F88 pin usages at LCD Board.....	64
6.2 LCD Board – USB Board communication protocol.....	66
7.1 PC side software development sources.....	69
7.2 DLL Methods.....	71
8.1 Experiment Results.....	85
B.1 Table for results.....	102

LIST OF FIGURES

FIGURE

2.1 QIC Mechatronics Boards: Processor Core and Carrier Board.....	14
2.2 QET DC Motor Control Board.....	15
2.3 Phidget Interface Kit 8/8/8 and Mini Joy Stick Sensor.....	16
2.4 Phidget Temperature Sensor - Thermocouple input and Phidget Accelerometer 3-Axis.....	16
2.5 EasyPic4 Board.....	17
2.6 Peripheral Boards by Mikroelektronika.....	18
2.7 NI ELVIS workbench and USB DAQ.....	19
2.8 Quanser Control Boards for NI ELVIS.....	20
2.9 General system.....	21
2.10 Peripheral side expansions.....	22
2.11 Computer side expansion.....	23
2.12 SPI single master multiple independent slaves connection diagram.....	29
2.13 Overall architecture.....	30
3.1 First design of USB Board.....	33
3.2 SMPS unit.....	33
3.3 USB Board and its layout.....	34
3.4 Schematic of USB Board.....	35
3.5 Pin numbing notation for Table 3.1 and Table 3.2.....	37
3.6 Flow diagram of USB Board firmware.....	39
4.1 Seven segment display driver.....	43
4.2 ULN2003 connection diagram.....	44
4.3 Parallel push buttons and dip switch.....	45
4.4 Linear voltage regulator.....	45
4.5 RS-232 communication unit.....	46
4.6 AIDIO Board and its layout.....	46

4.7 Schematic of AIDIO Board.....	47
4.8 Flow diagram of AIDIO Board firmware.....	50
5.1 Motor driver unit.....	53
5.2 Encoder interface.....	54
5.3 DAC unit.....	54
5.4 Motor Board and its layout.....	55
5.5 Schematic of Motor Board.....	56
5.6 Flow diagram of Motor Board firmware.....	59
6.1 LCD connection details.....	62
6.2 LCD Board and its layout.....	62
6.3 Schematic of LCD Board.....	63
6.4 Flow diagram of LCD Board firmware.....	65
7.1 Class Diagram of the DLL.....	70
7.2 Sample GUI.....	75
7.3 Port related controls of GUI.....	76
7.4 Input related controls of GUI.....	77
7.5 Output related controls of GUI.....	77
8.1 Experiment Setup.....	79
8.2 Sensors and actuators of the experiment setup.....	80
8.3 Encoded paper representing hard disk.....	81
8.4 Connections.....	83
8.5 Experiment GUI.....	84
B.1 Experiment setup.....	94
B.2 Reflective optical sensor CNY 70 and its application circuit.....	95
B.3 Interfacing circuitry of CNY 70 to Schmitt trigger integrated circuit	96
B.4 AIDIO Board interfacing.....	97
B.5 Adding reference.....	98
B.6 Initialization of DLL object.....	98
B.7 USB Board interfacing.....	100
B.8 Encoded paper.....	101

CHAPTER 1

INTRODUCTION

There is always a demand for more sophisticated, more intelligent products in the market. Engineers, with the continuous progresses in technology, have become capable of designing such products. The more sophisticated the products become, the more engineering disciplines should be included in the project. Mechatronics engineering is one of the multidisciplinary engineering, which is composed of mechanical, electrical, computational fields, has a major role in designing new products.

In mechatronics, there is an information flow between various mediums. As the name implies, these mediums basically are mechanical and electrical. Generally, the electrical medium is the intelligence intense part of the system and the mechanical part is for the interaction between this intelligence with the world which system will interact with. Mechanical systems might be controlled by microcontroller units (MCUs) or interfaced via MCUs to more powerful MCUs and/or computers. According to system constraints; flexibility, speed, usability etc; control medium is selected.

As seen from mechatronics design versatility, education of mechatronics engineering has to cover all these medium aspects; from high level computer programming to low level MCU programming. There are three trends at mechatronic education: low level paradigm focuses on low level interfacing like MCU controls; high level paradigm focuses on high level interfacing focuses on PC programming and hybrid approach combination of these paradigms. At [Timmerman] more information can be found about mechatronic education's low, high and hybrid level paradigms.

In Middle East Technical University (METU) Mechanical Engineering Department there are several mechatronic related courses which focus on high level programming and also there are courses focusing on interfacing, low level programming. And there is no standard hardware and/or software for these courses. So In this thesis control boards and computer programs are planned to be designed which allows users to control general mechatronic systems at low level and high level. In addition, another driving force is making a low cost, more affordable hardware so that students may purchase and use it at other projects of themselves. With this attempt, the students' interest might have been kept high at mechatronic field.

The information regarding the mechatronic courses around the world and METU will be presented in the following section. This information has been used as design constraints in the design of this thesis of which is discussed in detail at next chapter.

1.1 Overview on Mechatronics Related Courses

The mechatronic courses with their contents available at the universities and institutes are going to be presented in this section. This information will cover the course descriptions, experiments, projects, equipments, and boards (if there is). The information presented according to the latest syllabuses of the courses. The resources used at these courses will be presented as a whole at the end of this section.

1.1.1 ME 220 - Introduction to Mechatronics (METU)

This course is sophomore level course which introduce mechatronics design concept by making practice on simple mechatronic instruments. Because of this

course is introductory level, low and high level interfacing is covered equally with 6 experiments carried out during the semester along with the lectures [ME220].

Experiment 1 is about basic circuit elements, leds, resistors, capacitors and multimeter. At experiment 2 basic low level MCU programming and interfacing of PIC 16F877 application circuit to PC via serial port is done. Experiment 3 is RC servo motor application with PIC 16f877's pulse width modulation (PWM) signal. Experiment 4 is range sensing with PIC 16F877. Experiment 5 is close loop control with Basic Stamp (BS) which is light seeking platform with light sensor, RC servo motor and BS. Experiment 6 is using Basic Stamp2 (BS2) from Matlab. The objective of this lab is to practice the use of BS2 board via the Matlab functions.

1.1.2 ME 407 - Mechanical Engineering Design (METU)

This course is a senior level capstone design course. At the end of the course students has to design and manufacture a different mechanical engineering product determined by instructors at each semester. Although this course is not directly related with mechatronics, assigned projects are generally being a mechatronics product. This requires low level MCU programming and high level PC programming at intermediate level. To give a general idea about this course projects, some selected projects given last semesters will be presented.

Polymorphic Robotic System: Design of a modular robotic system consists of identical robots which interact with each other. Roller Coaster Simulator: Construction of hydraulically controlled Stewart platform. According to motion of platform related movie or pictures should be shown to users synchronously.

Military Mobile Robot: In this project a mobile robot platform is designed which could stand falling from 1 meter height and should be light as possible as for users can carry easily. There should be camera on the robot. Control of robot and data

transmissions from camera will be done wirelessly by a remote PC at distance up to 20 meter.

Intelligent Floor Vacuum Cleaner: Designing of vacuum cleaner which can traverse automatically the place where it is placed randomly and clean the room. It should detect walls, obstacles, and stairs and avoid them. It should be cordless.

1.1.3 ME 461 - Mechatronics Components and Instrument (METU)

ME 461 is an advanced version of ME 220 except it is senior level course covering content in more details. Like ME 220 low and high level interfacing is covered equally at this course.

The first three experiments are expanded version of ME 220's first two experiments. Experiment 4 is serial port monitoring, getting practice at PIC 16F877, PC serial interfacing. Experiment 5 is controlling of RC servo motor from PC via PIC 16F877's PWM signal. Experiment 6 is DC motor control with L293D which is H-bridge motor driver. Experiment 7 is step motors and their control. In this experiment a bipolar 4 wire step motor is controlled from PC.

1.1.4 ME 534 Computer Control of Machines (METU)

This course is graduate level course focuses on discrete control theory, high frequency systems; makes low level interfacing more important than high level interfacing implementations [ME534].

Project 1 is logic circuit design; 2 bit comparator is constructed by using only TTL ICs. Four switch inputs will be compared and according logic table related two led will be lighted. Project 2 is about sequential logic which is traffic light controller design by using TTL ICs and timer ICs. At project 3 automatic washer machine

(AWM) controller is done. AWM is simulated by hardware and software and controlled by PIC 16f877, TTL ICs.

1.1.5 ME307 - Mechatronics and Measurement Systems (Colorado State University)

This course is junior level course focusing on basic and advanced mechatronic instruments introductions and usages from diodes to data acquisition boards [ME307, Alciatore]. This content is covered in a hybrid paradigm.

Experiment 1 to 4 is about basics: resistor codes, breadboard, connectors, power supply, multimeter, oscilloscope, function generator, diodes. And also at experiment 4 a bandwidth filter is constructed by resistor-capacitor circuits, and presented its characteristics by function generator and oscilloscope. Experiment 5 is about switches and transistors. And a circuitry is constructed for relay, transistor comparison. Experiment 6 is about operational amplifier (Op-Amp) circuits. Characteristics of Op-Amps at different frequencies are shown by the constructed circuitry via function generator and oscilloscope. At Experiment 7 and 8 digital circuits are introduced. ICs and their manufacturing process stages are presented. And also a circuitry is constructed which is decade counter and seven segment display circuit driven by a 555 timer. Experiment 9 to 11 is PIC Microcontroller programming and interfacing. At these experiments PIC 16F88 is used. At these experiments content hexadecimal counter is done by flip-flops and by software. Seven segment display is used for display the result. And another experiment at these experiments is PWM speed control of a DC motor. Experiment 12 is about data acquisition. National Instrument USB data acquisition board is using with Lab View (PC software). Experiment 13 is strain gage rosette experiment. The weight of the mass on the simply supported beam is determined by reading strain gages. Experiment 14 is vibration measurement of bearings with piezoelectric accelerometer. A shaft is driven by motor and its speed is measuring by photo optic sensor by Schmitt trigger.

And also there are projects at the end of the course, done by groups. "Each group must design, build, test, and demonstrate a device controlled by a PIC microcontroller"[ME307]. Students propose their projects and it is graded according to six functional elements. These elements are:

- Output Display: Led, 7-segment digit display, LCD
 - Audio Output Device: buzzer, speaker with digitally pre-recorded music or voice, speaker with software-generated sound effects, speaker with software-controlled synthesized music or voice
 - Manual Data Input: switch, button, potentiometer, joystick, keypad, keyboard
 - Automatic Sensor Input: switch, photo-optic pair, potentiometer, photo cell, temperature sensor, encoder
 - Actuators: solenoid, on-off dc motor, PWM speed-controlled motor, reversible dc motor, stepper motor (unipolar, bipolar), closed-loop feedback servo motor
 - Logic, Counting, Integration, and Control: on-off motion control in one direction only, counting, programmed logic, motion in different directions and of different magnitudes, A/D and D/A interfaces, open-loop control, menu-driven software, advanced and/or multiple PIC microcontrollers, closed-loop feedback control.
- [ME307, Alciatore]

1.1.6 ME4447 - Microprocessor Control of Manufacturing Systems / ME6405 - Introduction to Mechatronics (Georgia Institute of Technology)

These two courses contents are slightly different from each other but sharing same laboratory work ME 4447 is senior, ME 6405 is graduate course. Courses are designed in a hybrid approach [Ume]. There are also student projects done in groups.

Experiment 1a is interfacing 68HC11 with a PC by evaluation board (EVB). In this experiment EVB is introduced. Serial communication is done by EVB's software. Experiment 1b is parallel output using leds. 4 leds are connected to the 4 IO ports of 68HC11. Sequentially leds are lighted. Experiment 2 A/D data acquisition by strain gauges. The strain gauge is mounted on a small aluminum cantilever beam. Voltage of strain gauge is amplified by an op-amp and is connected to A/D conversion ports of 68HC11. Experiment 3 PWM control of a DC motor using interrupts. A small DC motor is driven by PWM signal generated by 68HC11 via opto-coupler circuitry. Feedback for speed control of motor is done reading opto-interrupter signal generated by motor's slotted shaft.

"The Following are typical student projects: Automated Guitar Player, Cash Dispenser, Intelligent Floor Cleaner, Plant Care System, Autonomous Search Vehicle, Input Shaper on a Simple Pendulum, An Automated Coffee Bean Selector, Intelligent Baby Monitor, Smart Alarm for a House, Simulation of Hybrid Electric Vehicle's Control, An Automated Card Dealer, Semi-Automated Shift System for GT Motor sports, Hobbyist's 3-Axis Mill, Fish Feeder and Lighting Control, Remote Access Home Controller, Two Dimensional Laser Imaging Systems".[GIT Lab]

In experiments Motorola 68HC11 MCU is been used. And this MCU is used with Axiom CME-11-E9 Evaluation Board. By this EVB it is possible to program the 68HC11 via serial port of PC with its software. This board presents connectors for serial port, LCD, keypad, SPI, MCU's IO ports, A/D reference ports, bus ports of MCU. And it also contains solderless breadboard

1.1.7 ENGG3400/ENGG3490 - Introduction to Mechatronics Systems Design (University of Guelph)

This course is sophomore level design course. This course focuses on high level interfacing, controls theories. After some introductory lectures and laboratories

about mechatronic instruments; system model analyzes, intelligent algorithms, graphical based programming is heavily discussed and practiced [ENGG3400].

Experiment 1 is about data collection with sensors. These sensors are thermistors, Infrared (IR) reflective photo sensor, sonar, GP2D12 IR range-finder. Experiment 2 is measuring speed-voltage and torque-speed curves of DC motors. Hacked RC servo motors are used as DC motors. Matlab Simulink model of DC motor is created and compared with experiment step input results. Experiment 3 is localization with a differential drive mobile robot base. Wall following behavior is implemented to robot by feedback of sonar, IR range finder and optical mouse combinations. Then robot tries to follow a circular path and its dead reckoning errors are determined compared with sensor readings. Experiment 4 is about obstacle detection of mobile robot platform. With range sensors (sonar, infrared, rangefinder) various obstacles (board, triangle, square box, rectangle box, circle) are tried to be detected in three minutes.

At this course, Handy Board with Expansion Board is using. The Handy Board is a board uses 52-pin Motorola MC68HC11 processor. It has 4 outputs for DC motors, a connector system that allows active sensors to be individually plugged into the board, an LCD screen.

The Expansion Board plugs on top of the Handy Board, and provides the following features: 10 additional analog sensor inputs; 4 inputs for active LEGO sensors (reflectance sensor and shaft encoder); 9 digital outputs; 6 servo motor control signals with power supply from the Handy Board's internal battery; optional external power for servo motors; connector mount for Polaroid 6500 ultrasonic ranging system; general-purpose electrical prototyping area; pass-through connector for the Handy Board's LCD screen. [HandyBoard]

1.1.8 151-0641-00 - Introduction to Robotics & Mechatronics (Institute of Robotics and Intelligent Systems)

This course is a freshmen level introductory course. So as an approach, high level paradigm is chosen. Experiments are carried on ready to use hardware and setups, high level programming is practiced [IRIS].

Experiment 1 is about programming to introduce the students to the systems that will be used for the course. Experiment 2 is data collection from joystick. Sensoray 626 card is used for data acquisition. Experiment 3 is infrared range finding sensor and its calibration. Experiment 4 is about sampling theorem and digital filtering. Experiment 5 is DC motor control and analyzing its transfer function. Experiment 6 is PID control of dc motor with models and sensor feedback. Experiments 7-9 are about controlling two link robotic arm. At the end experiments robot arm draw shapes.

At this course Sensoray 626 card is been using. This card's features are; 4 14-bit D/A outputs, 16 16-bit differential A/D inputs, 6 versatile 24-bit counters for encoders/timers with interrupt generation, 48 digital IO, having xPC target real-time kernel. Card is connected to PC via PCI bus.

1.1.9 2.737 - Mechatronics (Massachusetts Institute of Technology)

This course is graduate level course focuses on low level control, interfacing of mechatronic instruments [MIT2.737, Trumper]

Experiment 1 is servomotor control. A DC motor with integral tachometer is driven digital to analog (DA) input with power amplifier. For speed feedback tachometer, for position feedback single turn pot is used. Experiment 2 is about analog feedback systems. In this experiment designing and stabilizing power op amp circuits is covered. Experiment 3 is digital electronics and encoder interface.

Constructing up/down counter by logic gates, binary up/down counter and binary-coded decimal to seven segment decoders interfacing is done.

At experiments dSPACE DS1102 controller is used. It has six 20 kHz PWM outputs, four 16 bit A/D inputs, four 16 bit A/D outputs, two incremental encoder inputs, 16 bit digital I/O. It is connected to PC via ISA bus. Graphical control algorithm development is done by using Matlab Simulink/Real time workshop.

1.1.10 ME 333 - Introduction to Mechatronics (Northwestern University)

This course is sophomore level course prepared in a hybrid approach. In laboratory work instruments that commonly used at their projects are practiced and at the end of the course with this knowledge a final project is designed and manufactured [ME333].

Experiment 1 is investigating The Furby mechatronics toy and driving its motor manually and reading its encoder signal by Schmitt trigger. Experiment 2 is about close loop control of motor. Op-amps are used for current amplifier and in feedback control of a motor. Experiment 3 is using Matlab xPC target as PC104 stack. Close loop control of a motor is done by encoder feedback. Experiment 4 is extended version of experiment 3; addition to it, digital input and output is examined at real time xPC target.

There are also projects at the content of the course. And these are the some selected projects: Automatic card dealer, the elevator, a controllable flexible beam, automatic leveling platform, XY plotter, laser skeet shooting.

The control hardware used in this course is PC104 based embedded platform with DAQ Sensoray 526 card. DAQ card has 8 analog-to-digital converters (ADC), 4

digital-to-analog converters (DAC), 8 digital input-output lines (DIO) and 4 quadrature decoders.

1.1.11 ME210 – Introduction to Mechatronics (Stanford University)

This course is senior graduate level course and lectured as ME 333 in a hybrid approach. Low level interfacing is covered at experiments, high level interfacing is covered at projects [ME210].

Experiment 1 is about laboratory environment, resources introduction. Experiment 2 is op amps and comparators and their characteristics. Experiment 3 is low level programming and implementation. Programming is done on Motorola 68HC11 MCU with MicroCore-11 Board. Experiment 4 is about speed control of DC and step motors. Speed is adjusted by a pot and motors are driven by PWM outputs. And also by H-bridge direction control is done.

At the experiments MicroCore-11 Board is using with 68HC11 MCU. It has eight channels 8 bit ADC, eight digital inputs, 2 PWM comparator channel, RS-232, SPI interfaces. And it connects other boards such as prototyping board, backplane to breadboard and motor driver board.

1.2 Result of Course Survey

The course survey aim is to discover the trend of the mechatronic education. Table 1.1 presents a summary of the conducted survey. Average usage of hardware resources listed in this table provides a hint to board designers about how much resources should be allocated with these cards. At the end of this survey, it is seen that all three paradigms (low, high and hybrid) are equally distributed. And standard control and/or sensor boards are widely used. The resources used by experiments are various. Differences occur according to the courses' aims. As a

result, this thesis tries to develop an embedded system for usage at METU courses presented above and courses choose hybrid paradigm for their course content.

Table 1.1 Overview of courses and experiments

Course levels	graduate	senior	junior	sophomore	freshmen
# of experiment	6	3	1	3	1
Experiment type	introduction to instruments	digital I/O manipulations	sensor measurement	motor control	
# of experiment	12	11	17	21	
	Avarage usage of				
Resources/ School	Analog inputs	Digital inputs	Analog outputs	Digital outputs	
Colorado State	1	3	1	7	
Georgia Tech	1	1	1	4	
Guelph	3	1	2	6	
IRIS	2	6	2	2	
METU	2	3	1	6	
MIT	3	2	1	14	
Northwestern University	1	1	1	1	
Ohio State	2	6	2	2	
Stanford	1	1	1	1	

CHAPTER 2

REVIEW OF THE STATE-OF-THE-ART

At previous chapter, the reasons for the requirement of a mechatronic board have been addressed. In this chapter, frames of the board(s) will be drawn; products at the market, the architecture, the design path will be presented.

2.1 Boards Available at the Market

There are a lot of boards at the market about mechatronic education and applications. Some of these products are development boards of specific MCU; some of them are dedicated boards for a specific usage. Some of them are designed for modular usage, while others are designed in a monolithic fashion. The price range of the boards suitable for educational use is also very wide. To give a general idea about the products available in the market, a set of manufacturer are selected and relevant products of them are discussed.

2.1.1 Quanser Products

Quanser has several products for mechatronic education and researches. Some of the boards come with experiment setups. [Quanser]

QIC Mechatronics is a development board for PIC 16 and 18 series. It consists of 2 types of board of which one is processor core board and the other is carrier board (Figure 2.1). Processor core board connects the top of the carrier board with pin headers. Processor board has 10 bit ADC with 8 channels and In-Circuit Serial Programming (ICSP) unit. High level programming environments are Matlab/Simulink and C/C++. Carrier boards properties are dual encoder channels,

24 configurable digital I/O, 8 PWM channels with 18V, 500mA, 2 PWM channel with H-bridges up to 24V, 3A

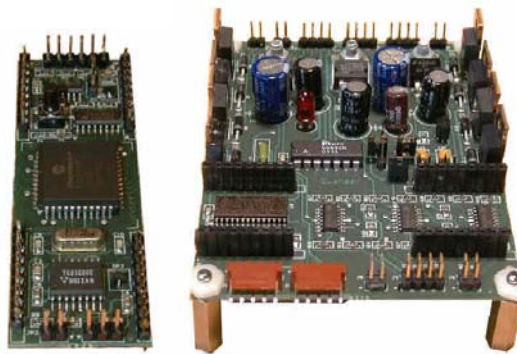


Figure 2.1 QIC Mechatronics Boards: Processor Core and Carrier Board [Quanser]

QET DC Motor Control Board (Figure 2.2) has DC brush motors with analog tachometer, quadrature encoder mounted on motor and feedback potentiometer. Control of this setup can be made by analog input from breadboard on the board, digital with external DAQ boards, or embedded PIC MCU on the board with RS-232 or USB1.1 communication.



Figure 2.2 QET DC Motor Control Board [Quanser]

2.1.2 Phidgets Products

Phidgets are little modular boards that work with only PCs via USB. Development environments are Visual Basic, VBA (Microsoft Access and Excel), VB.Net, C#, LabView, Matlab, Java, Delphi, C and C++. [Phidgets]

Interface Kits Boards are used for analog inputs and digital I/Os. There are 0/16/16, 8/8/8, 0/0/4 types; for example 0-16-16 means that the board has 0 analog inputs, 16 digital inputs, 16 digital outputs. These I/Os are accessed via connectors from other boards. There are analog sensor boards which are IR distance sensor, IR reflective sensor, vibration sensor, light sensor, force sensor, humidity sensor, magnetic sensor, rotation sensor, touch sensor, motion sensor, slider, mini joystick sensor, temperature sensor, pressure sensor, multi turn rotary sensor, voltage sensor, 20A or 50A current sensor. These analog sensor boards only connect to interface boards with 3 pins. 8/8/8 boards also contain a 2 port USB Hub which makes other types of USB boards connect via interface boards (Figure 2.3).

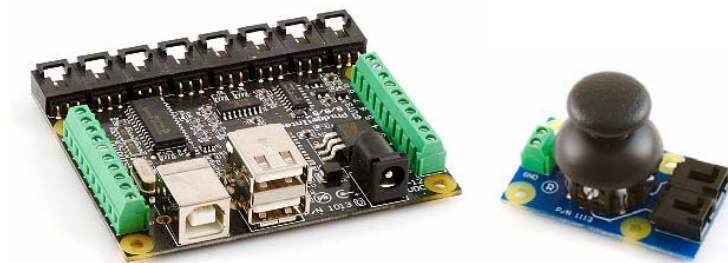


Figure 2.3 Phidget Interface Kit 8/8/8 and Mini Joy Stick Sensor [Phidgets]

LCD/LED boards are modified version of 8/8/8 interface kit which a 20x2 LCD is integrated at the back of the board. The LED board is capable of driving 64 leds. For motor control applications, there are two types of boards. One is for RC servo motor applications and the other is for driving DC motors. There are two types of RC servo motor boards. One board can drive only one motor whereas the other can drive up to 4. USB Sensors boards are sensor integrated USB boards. They are temperature sensor-thermocouple, encoder, high speed encoder, accelerometer, 3 axis accelerometer, pH sensor (Figure 2.4)

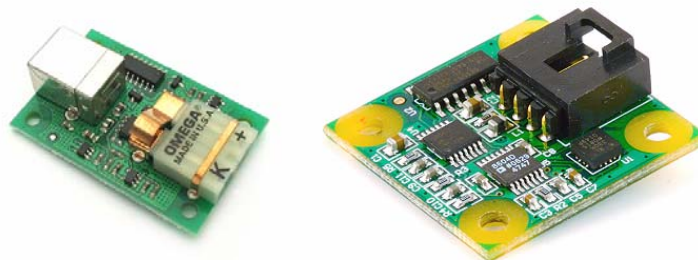


Figure 2.4 Phidget Temperature Sensor - Thermocouple input and Phidget Accelerometer 3-Axis [Phidgets]

2.1.3 Mikroelektronika Products

This firm's products are consists of development boards for several MCUs with extra controller boards and compilers. They supply a rich library of example codes written as well. [Mikroelektronika]

For example EasyPic4 board (Figure 2.5) is a development board for 8, 14, 18, 20, 28 and 40 pin PIC MCUs with in circuit programmer and debugger. It is programmable over USB interface. Power supply can be selected between USB and external source. On the board there are 36 leds, 36 buttons, 2 pots, 1 LCD, 1 graphic LCD (GLCD) , 4 7 segments, 1 RS-232 connector, 1 USB connector for MCUs capable of USB communication, pin headers for external usage, a digital thermometer.



Figure 2.5 EasyPIC4 Board [Mikroelektronika]

EasyPIC4 board serves a rich set of resources to a user which makes it a large board. Smaller boards are also offered by this company such as real time clock board, digital analog converter (DAC) board, CAN bus board, IR communication board, EEPROM board, and keypad board (Figure 2.6).



Figure 2.6 Peripheral Boards by Mikroelektronika [Mikroelektronika]

There are also similar products of Mikroelektronika for Microchip dsPic, Cypress PSoc, Atmel AVR, Motorola HC908 MCUs. The idea is same for these boards; there are a lot of leds, buttons, several sensors, connectors with USB flash programmer.

2.1.4 National Instruments Products

National Instruments has several types of product at instrumentation and measurement fields. For academic usage they developed a workbench with several add-ons which its name is Educational Laboratory Virtual Instrumentation Suite (ELVIS). This product simulates several lab equipments like digital multimeter, variable power supply, oscilloscope, function generator, bode analyzer, dynamic signal analyzer. Also there is an USB DAQ device of NI for analog digital input

output and for PC interfacing. These products come with software called LabVIEW graphical programming tool which is used for configuring and programming the NI devices.

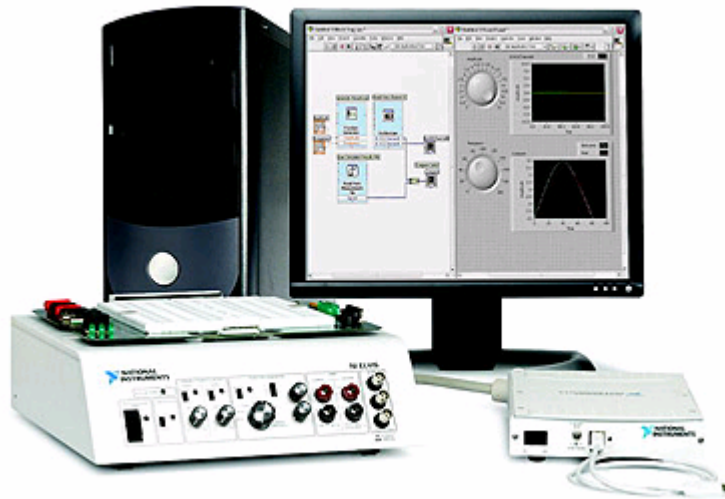


Figure 2.7 NI ELVIS workbench and USB DAQ [NI]

There is a prototyping area at the top of the workbench which is detachable. There are companion products of used with ELVIS. Quanser has control boards designed for ELVIS like DC Motor Bundle, DC Rotary Inverted Pendulum Bundle and HVAC Trainer Bundle (Figure 2.8). There is a microcontroller platform version of ELVIS using Freescale HCS12 MCU.



Figure 2.8 Quanser Control Boards for NI ELVIS [NI]

2.1.5 Result of Market Survey

Using a product from the market for mechatronic education and research at ME department of METU has some drawbacks. The main disadvantages are cost, availability and maintainability. As these products are generally imported products, if the products become malfunctioned, this increases their expense even further. The firmware accessibility could be limited for modifications if not completely inaccessible. Also technological improvements shorten this kind of products' lifetimes. According to course curriculum sometimes modifications has to be made on these boards which necessitates fully transparent product architecture. Designing your own board(s) provides this by making users more independent in modifications. So, lectures could dictate what kind of board(s) needed, not vice versa. As a result, instead of building a system top of the product on the market, designing custom board(s) is preferred.

2.2 Architecture

An interfacing medium is planned to be designed and manufactured as a part of this thesis work. There will be high level control medium such as computer and a physical medium which is the mechatronic system to be controlled / interacted with (Figure 2.7).

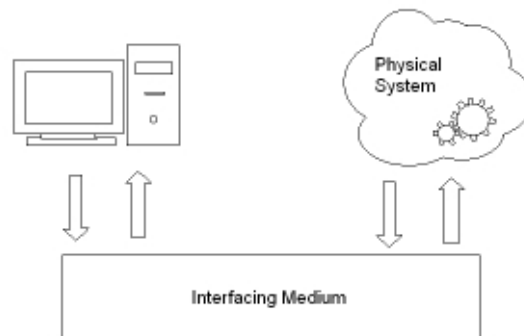


Figure 2.9 General system

At course and market survey, it is seen that physical systems are various. There is not a single module that handles all kinds of IOs. So first, the kind of modules which should be used for such systems is determined. In the second phase, combination of these modules is determined. At this point, combination could be done either as monolithic or modular paradigm. Modular approach is selected for future developments. So there will be a board which communicates with PC and when system demand increases several other boards which will be connected to this board in a modular fashion.

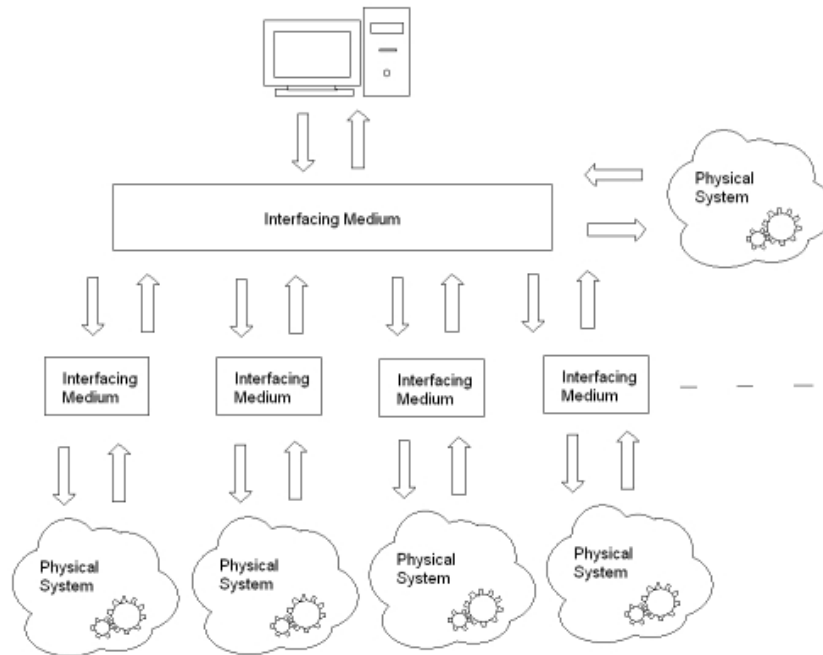


Figure 2.10 Peripheral side expansions

If the overall system demand is still much more than the designed system shown in the Figure 2.8 computer side interfacing should be available for expansion (Figure 2.9).

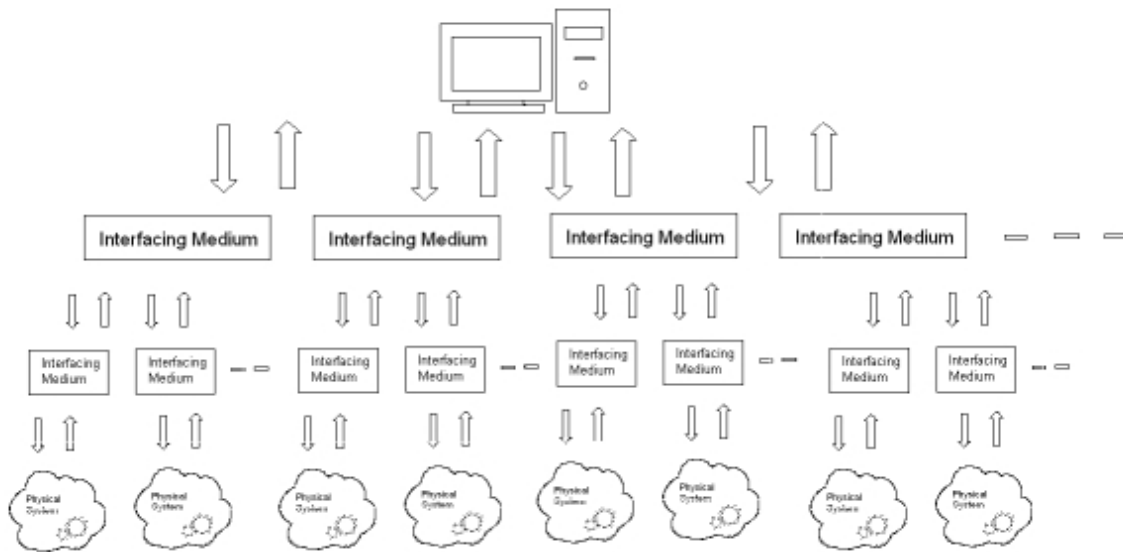


Figure 2.11 Computer side expansions

2.2.1 Computer Interfacing

Every communication protocols and standards has pros and cons. But at least; ease of usability, being hot pluggable, fast enough for usage, room for expandability fulfill the needs of PC communication.

Potential interfacing standards will be discussed in the below and one of them will be chosen for design criteria.

2.2.1.1 Serial Port

Serial port is an interface on a computer system with which information is transferred serially means in or out one bit at a time; this was accomplished using the RS-232 standard over simple cables. Originally 25 pin D-type connector was specified but later 9 pin D-type connector is replaced. The RS 232 standard is

formally limited to 20 Kb/s; serial ports on popular personal computers allow settings up to 115 Kb/s. Transmission may be synchronous or asynchronous. By the developments in technology, serial port is supplanted by newer standards such as USB and FireWire.

2.2.1.2 Parallel Port

At Parallel port, data is transferred in or out in parallel fashion, that is, on more than one wire. A parallel port carries one bit on each wire thus multiplying the transfer rate obtainable over a single cable. It has two modes EPP and ECP with 1.5 Mb/s and 2.5 Mb/s speed respectively. On many modern computers, the parallel port is omitted for cost savings, and is considered to be a legacy port.

2.2.1.3 USB

USB is a serial communication protocol designed for variety of peripherals from mouse to video cameras to share a standard connection. There are hosts located at computers or at devices and slaves which are devices, peripherals and up to 127 devices may be connected to a single host controller. USB connectors also carries power lines (5V, 500mA) which makes some devices work with only USB cables. USB devices are hot pluggable that means after enumeration of devices and determination of driver, there is no need for rebooting the computer. It has 3 types of devices which determine data transfer rates; low speed, full speed and high speed devices at 1.5 Mb/s, 12 Mb/s, and 480 Mb/s transfer speeds respectively.

2.2.1.4 FireWire

FireWire is a personal computer and digital video serial bus interface standard offering high-speed communications and isochronous real-time data services. FireWire can be considered as a successor technology to the obsolescent SCSI

Parallel interface [W_FireWire]. Up to 63 devices can be daisy-chained to one FireWire port. It allows peer to peer communication between devices which uses less system CPU memory but also makes it more costly than their USB equivalents. FireWire devices are plug and play and hot pluggable devices. It has 2 versions FireWire 400 and FireWire 800 whose transfer rates are 400 Mb/s and 800 Mb/s respectively.

2.2.1.5 PCI Express

PCI Express (PCIe) is a development of the old technology PCI (Peripheral Component Interconnect) bus. Unlike PCI, PCIe is a serial bus. Its topology is like point to point network topology. All PCIe devices connected to motherboard processor which is also integrated PCIe switches, at separate connections. Communication is real time with handshaking and error detection. There are several types of PCIe devices according to their physical connections also called links from 1 to 32. One bit transfer is done at one clock cycle at one lane and 8 GB/s maximum transfer rate for 32 link device. PCIe devices might be used as hot-pluggable according to hardware.

2.2.1.7 Comparison and Selection

Serial and parallel ports are seen as legacy ports like PCI slots. Today computers are composed of USB, FireWire, PCIe busses.

When we compare USB and FireWire; USB can be used as low and medium bandwidth usage and FireWire for high bandwidth. They probably coexist in the computers instead of legacy ports. But for our case (communication with mechatronic system) USB is sufficient and using FireWire will be unnecessary investment. FireWire has controller chip at every device; on the other hand USB has one controller chip on the computer which makes USB device development

cheaper than FireWire. Besides, FireWire still isn't a popular interface at computers. On the contrary, at least 2 USB ports are exist at even old computers.

Table 2.1 Overview of interfaces

interface	speed	max device
serial	115 Kb/s	1 per slot
parallel	8 Mb/s	1 per slot
USB	1.5, 12, 480 Mb/s	127
Firewire	400, 800 Mb/s	63
PCIe	250 MB/s per lane	1 per slot

PCIe seems a very attractive solution for interfacing mechatronic systems. But it has some disadvantages for usage. Firstly, it is not easy to plug or unplug in comparison to USB devices. And secondly, it became popular recently, so it requires some time to be a standard (at least for our lab computers). But definitely, this interface should be used for mechatronic devices in the future. Consequently, in this thesis USB is selected for the reasons that are mentioned above.

Whether it is USB, or FireWire, or PCIe, the main idea of these products was originally created to standardize external connectors. However, neither product can replace another because of their different characteristics and application. Each product has its own unique function.

The USB port is replacing all the old serial and parallel ports as the first choice method for attaching peripherals. The reasons are listed below:

- Hot pluggable and configurable

- It is fast enough (up to 480 Mb/s)
- Flexible for different data transfer types
- Safe with error detection mechanisms
- It works in real time, this makes it ideal for control or monitoring applications where the data is used immediately
- It has its power supply, thus an external supply for the device may not be required. This means fewer wires and greater portability.
- Popular on computers
- Expandable up to 127 devices

When our demands are considered, USB seems to be a good start. For future work PCIe interface should also be developed for mechatronic systems.

2.2.2 Peripheral Boards

As seen from the lectures and market products, the board should be modular for scalability and connectivity that makes it more robust and usable. These properties also determine the number of boards. All these types of boards should be not only interconnected but also should be connected to PC independently. In addition to these, a new type of board architecture could allow expandability.

For the present, there are 3 types of board designed to allow expandability: analog input digital I/O board, motor control board, display unit board. Before giving more details, other design criteria will be discussed regarding the interfacing of peripheral boards.

2.2.3 Peripheral Interfacing

Peripheral side interfacing is one of the important issues that have to be mentioned. There are two major bus standards for inter peripheral communication:

inter integrated circuit (I2C) and serial peripheral interface (SPI). These two standards, that both work as master slave combinations, are supported by integrated circuits, Electrically Erasable Programmable Read-Only Memory (EEPROM) devices and sensors.

2.2.3.1 I2C

There are two wires which are pulled up with resistors for I2C communication. One is serial clock the other is serial data wire. I2C is a multiple master multiple slave communication standard. Communication is handled by a protocol. First master(s) publish which slaves it wants to exchange data. Then determines data flow direction (read or write operation) There are acknowledgement bits to start and to stop the communication. Because of slave addressing usage of I2C is ideal for multiple slaves, but makes it a bit slower as 3.4 Mbit/s for new I2C standard.

2.2.3.2 SPI

At SPI communication, three or four wire is being used. This standard works like shift registers. One wire for serial clock, one wire for serial data input, one for serial data output and one for slave select of which is optional according to multiple slave applications. SPI is working at full duplex mode and ideal for single master single slave usage. There is no protocol usage at default, because master communicates only one slave at a time. As a result of working at full duplex mode without acknowledgement, SPI is addressing its maximum speed at 10 Mbit/s which is faster than I2C. In general use; first master sends its data then waits for slave's data so data exchange is done. Multiple slave application is done via cascading slaves in a daisy chain fashion or using slave (or chip) select wire which sends active low signal to the slave that will be on the bus for exchanging data at the moment.

2.2.3.3 Selection

The selection is made mostly according to speed and simplicity criterions. As a result, SPI is selected for peripheral communication. When it comes to the type of multiple slave connection configuration; they can be connected to the Master in parallel (independently enabled slaves) or in series (daisy-chained slaves). In a daisy-chained configuration, the limit of slaves can be increased without requiring any additional resources from the master on the expense of added communication delays. In this configuration, when a slave device is added or removed from the chain the necessary connection modification is needed for continuity of chain, whereas in parallel configuration no such modification is necessary. Considering the maximum amount of resources that might be required in mechatronic experiments, slave select is an acceptable solution. A closer analysis reveals that up to a maximum of 4 slaves will be enough to conduct all the necessary lab studies planned. Considering the fact that, parallel configuration not only makes addition and removal of slaves easier, but it also provides faster access times between the master and the slaves. Finally, in parallel configuration slaves does not have to support daisy chaining at all. This might make custom board design easier and also enable the usage of SPI enabled devices that can only work in parallel configuration. As a result, independent slaves with slave (or chip) select (parallel configuration) is preferred for this design for the sake of modularity and usability. A sample configuration is illustrated in Figure 2.10.

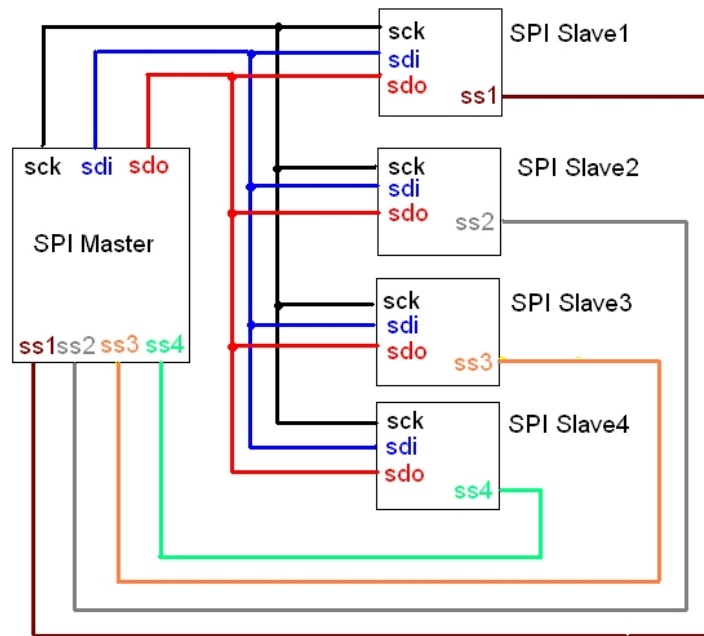


Figure 2.12 SPI single master multiple independent slaves connection diagram

2.3 Conclusion

From the literature survey and design criterions a scalable, modular architecture is designed (Figure 2.11). There is a USB Board that communicates with PC via USB. From the feature of USB standard, it is possible that 128 USB Board can be connected. For complementary usage to USB Board, there will be other boards responsible for analog digital I/O, motor control, LCD driving boards. These boards are interfaced to USB Board via SPI. Hence, USB Board is master for SPI communication. The other boards are independent and can be connected to any SPI connectors. All the peripheral information can be collected at USB Board or at the PC.

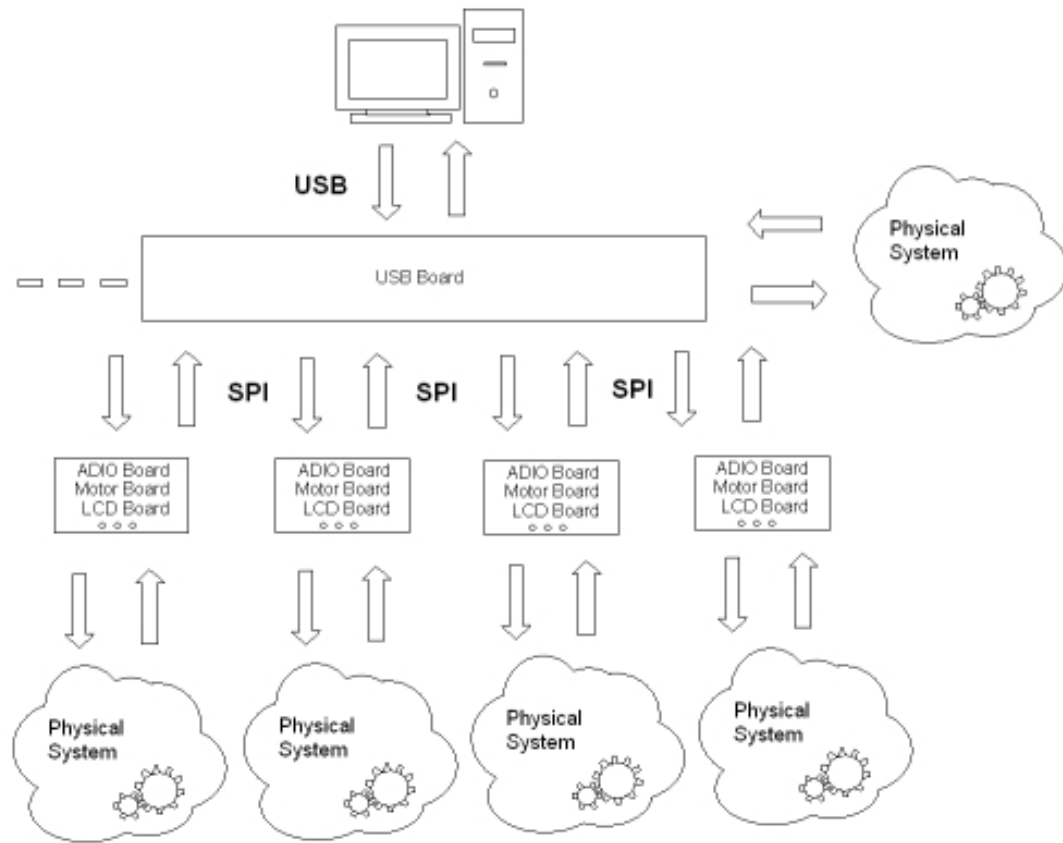


Figure 2.13 Overall architecture

CHAPTER 3

USB BOARD

In the concept of this thesis, several boards aforementioned are designed, manufactured and coded. This chapter will cover in depth explanation of the USB Board regarding its function at the frame of overall system, its hardware and firmware. A small revision history of this board will be also given. The other details like circuit design, firmware code information will be presented at appendices.

3.1 Design Stages and Hardware

The design process of this board is started at the selection of MCU. Selection is made according to the below criterions:

- USB speed of chip
- Chip's features, resources
- Availability of chip
- Availability of its development accessories
- Popularity

USB standards define three types of USB devices according to data transfer speed: low, full and high speed devices. Their speeds are 1.5, 12, 480 Mbit/s, respectively. A low speed device design is eliminated because of its speed which is under expectations for an embedded system. At high speed MCU market, the application fields are generally focused on mass storage applications. So, generally these chips don't have so much digital analog inputs- outputs, and have direct memory access feature for communication with other peripherals especially for memory cards. And the chips used at USB high speed data acquisition boards are

generally produced as application specific integrated circuit fashion. As a result, full speed chips are examined which is more popular for general IO applications. There are many products at full speed USB market. After eliminating the products according to their features and resources, Cypress, FTDI and Microchip companies are remained. Among these brands, Microchip was selected by considering the criterions of availability of these chips and their development tools in Turkey, as well as their usage popularity in the world. At last, the chip model 18F4550 of Microchip is selected as being the newest product of this brand in this product field at that time.

At the early stages of design, this board is tried to be made as compact as it can be. Only two leds (one is for USB power status, one is for general usage) and two push buttons (one is for reset, one is for general usage and boot loader) are put on the board and all the other pins are carried to external usage via male pin headers at the bottom of the board (Figure 3.1). By this way, its interfacing to breadboard became easy. Resources and features on the chip are thought to be sufficient for the concept of this thesis. If the physical system control demand exceeds the resources of one board, it was planned to use several of boards by the help of USB scalability. Then it is understood from literature survey and the nature of the problem that one type of an USB board will not be enough. At this level, two design ideas are discussed. One is a board design with all the components on at one board with a USB communication capability. The other is one USB communication board with complementary boards. The second idea is decided for the sake of modularity and future developments.

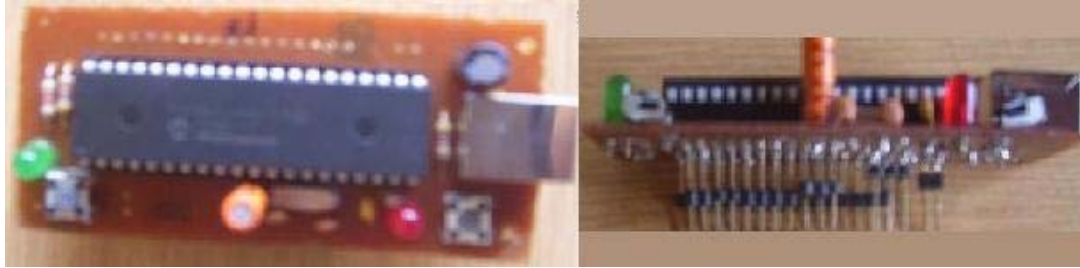


Figure 3.1 First design of USB Board

After selection of SPI for peripheral communication, the first design evolved as the needs of peripherals. And architecture leads to that all the peripherals will be connected to USB Board. So a power supply unit is also added to USB Board for supplying power for peripheral MCUs via peripheral communication cables (Figure 3.2 and 3.3). A switched mode power supply (SMPS) is selected as the power supply unit. The SMPS unit's maximum input voltage is 40V and output voltage is 5V with supplying current up to 3A loads.

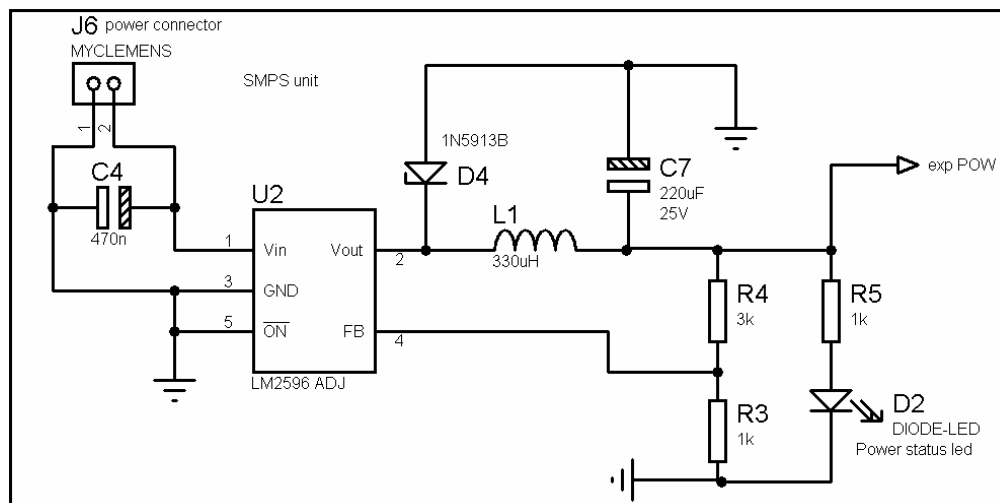


Figure 3.2 SMPS unit

Finally, a SMPS unit, four ribbon cable connectors are added to first design. For standard cable usage, the male pin header for unused pins is replaced by two ribbon cable connectors. SMPS and USB power is also carried for these connectors. A tricolor led is replaced instead of one led, USB power status led is preserved and SMPS power status led is added. Two push buttons are also preserved (Figure 3.3 and 3.4).

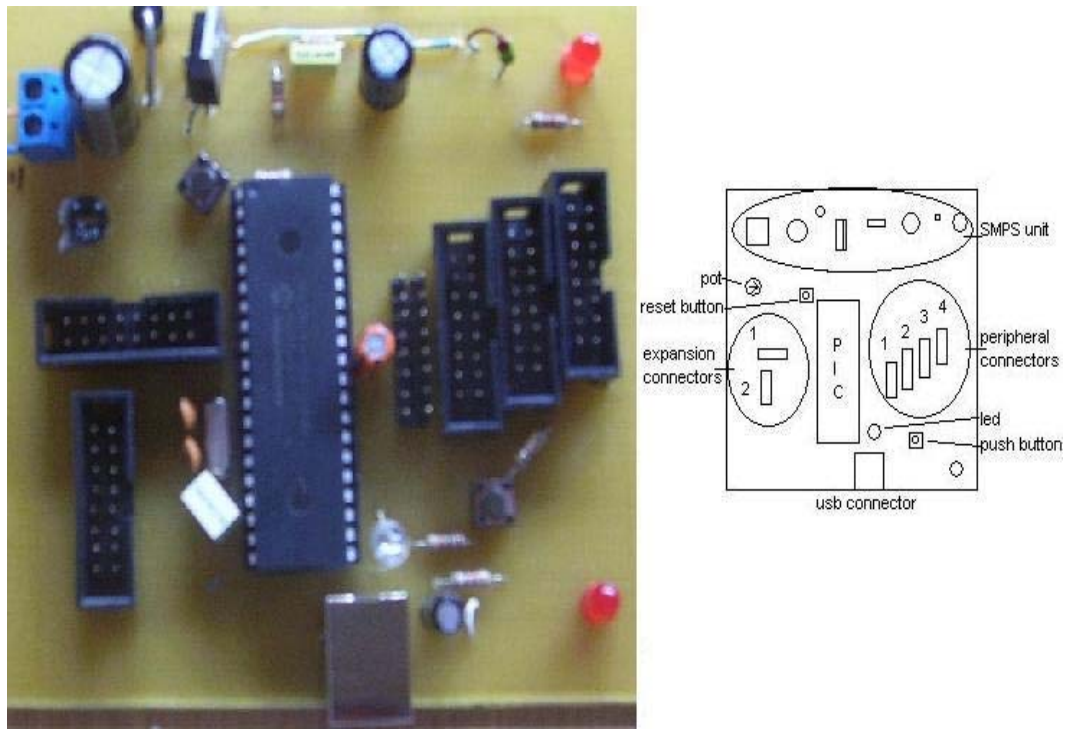


Figure 3.3 USB Board and its layout

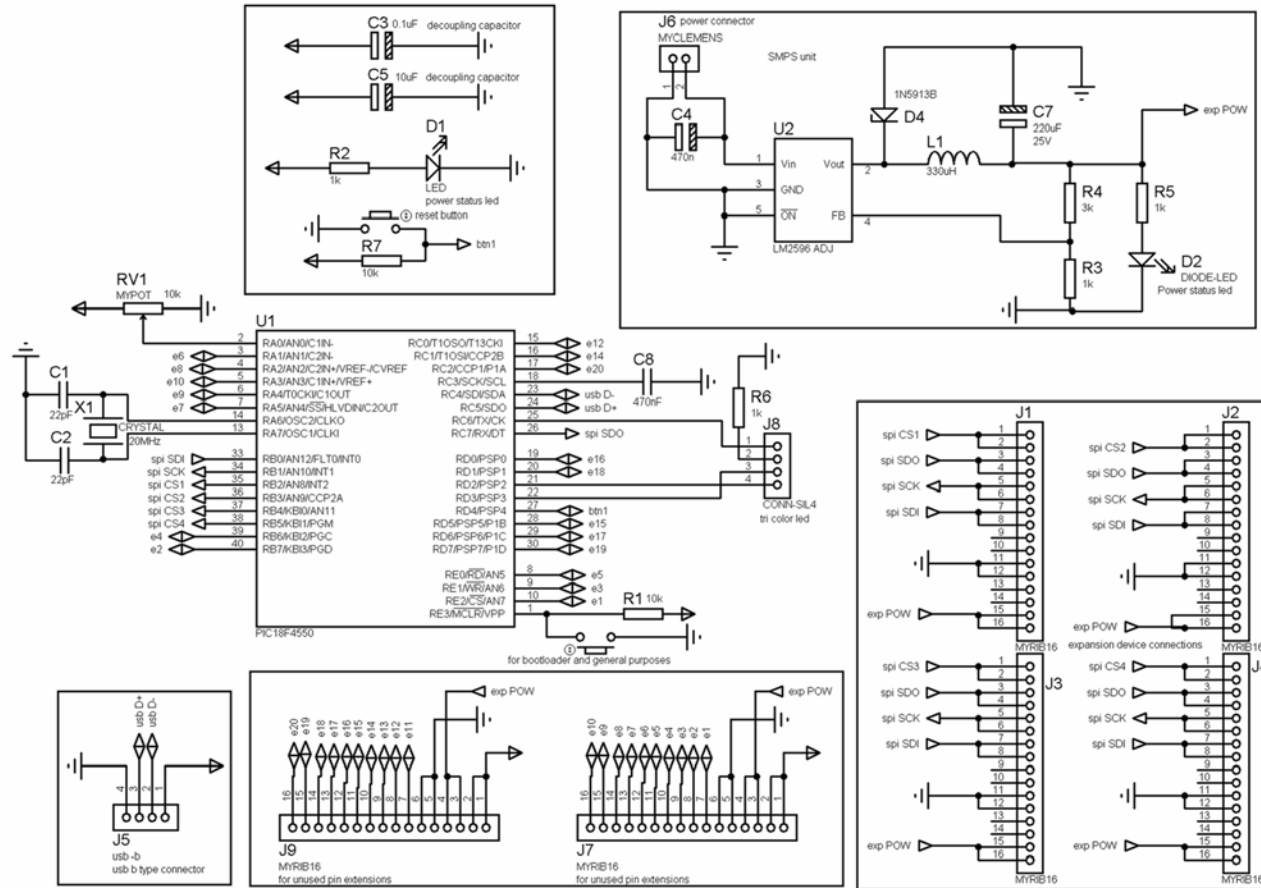


Figure 3.4 Schematic of USB Board

The connection details of MCU and ribbon cable connectors are presented in Table 3.1, 3.2 and also shown at Figure 3.4. Pin number notations are shown in Figure 3.5.

Table 3.1 PIC 18F4550 pin usages at USB Board

PIC 18F4550 pins usage					
description	connected to	pin #	pin #	connected to	description
resetting system	reset push button	1	40	expansion ribbon cable connector 1	external general usage
general usage	pot 1	2	39	expansion ribbon cable connector 1	external general usage
external general usage	expansion ribbon cable connector 1	3	38	peripheral ribbon cable connector 4	spi CS 4
external general usage	expansion ribbon cable connector 1	4	37	peripheral ribbon cable connector 3	spi CS 3
external general usage	expansion ribbon cable connector 1	5	36	peripheral ribbon cable connector 2	spi CS 2
external general usage	expansion ribbon cable connector 1	6	35	peripheral ribbon cable connector 1	spi CS 1
external general usage	expansion ribbon cable connector 1	7	34	peripheral ribbon cable connectors	spi SCK
external general usage	expansion ribbon cable connector 1	8	33	peripheral ribbon cable connectors	spi SDI
external general usage	expansion ribbon cable connector 1	9	32	supply voltage from USB	
external general usage	expansion ribbon cable connector 1	10	31	ground	
	ground	11	30	expansion ribbon cable connector 2	external general usage
	supply voltage from USB	12	29	expansion ribbon cable connector 2	external general usage
	oscillator	13	28	expansion ribbon cable connector 2	external general usage
	oscillator	14	27	push button 1	general & boot loader usage
external general usage	expansion ribbon cable connector 2	15	26	peripheral ribbon cable connectors	spi SDO
external general usage	expansion ribbon cable connector 2	16	25	tri-colored led	general usage
external general usage	expansion ribbon cable connector 2	17	24	USB connector	USB data D+
	stabilizer capacitor	18	23	USB connector	USB data D -
external general usage	expansion ribbon cable connector 2	19	22	tri-colored led	general usage
external general usage	expansion ribbon cable connector 2	20	21	tri-colored led	general usage

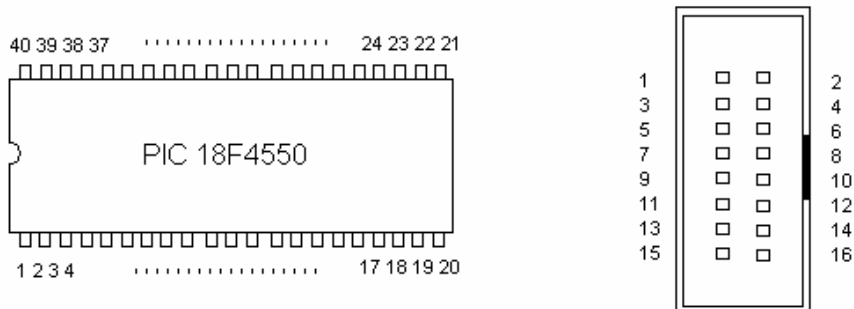


Figure 3.5 Pin numbering notations for Table 3.1 and Table 3.2

Table 3.2 Ribbon cable connectors' pins' descriptions

connected to	pin #	pin #	connected to
peripheral ribbon cable connector #X			
PIC spi CS X	1	2	PIC spi CS X
PIC spi SDO	3	4	PIC spi SDO
PIC spi SCK	5	6	PIC spi SCK
PIC spi SDI	7	8	PIC spi SDI
unused	9	10	unused
ground	11	12	ground
unused	13	14	unused
supply voltage from SMPS unit	15	16	supply voltage from SMPS unit
expansion ribbon cable connector 1			
supply voltage from USB	1	2	supply voltage from USB
supply voltage from SMPS unit	3	4	supply voltage from SMPS unit
ground	5	6	ground
PIC RE2/AN7	7	8	PIC RB7
PIC RE1/AN6	9	10	PIC RB6
PIC RE0/AN5	11	12	PIC RA1/AN1
PIC RA5/AN4	13	14	PIC RA2/AN2
PIC RA4	15	16	PIC RA3/AN3
expansion ribbon cable connector 2			
supply voltage from USB	1	2	supply voltage from USB
supply voltage from SMPS unit	3	4	supply voltage from SMPS unit
ground	5	6	ground
unused	7	8	PIC RC0
unused	9	10	PIC RC1/CCP2
PIC RD5	11	12	PIC RD0
PIC RD6	13	14	PIC RD1
PIC RD7	15	16	PIC RC2/CCP1

3.2 Firmware

There are two types of firmware on the chip. One is boot loader firmware the other is application firmware. Boot loader firmware is written on the flash memory of chip and it is enabled by interrupt routines. When boot loader instruction sets are executed it enables users to upload their application code to MCU via PC by a serial interface (in this case it is USB). But first a boot loader code is needed to be uploaded to MCU via its programmer hardware. Then to use the boot loader code one should refer to the usage of boot loader at application code as first instruction. If this isn't done, boot loader code will be erased and programming by hardware should have to be redone.

Microchip' supplied code is used as boot loader firmware. According to this code, if the pin 37(port B4) is at logic low state, the boot loader is activated during start up. At development hardware of 18F4550, a push button is connected to this pin. This firmware is modified according to the designed circuitry. The push button is connected to pin 27(port D4). To enter the boot loader mode, push button at pin 27 is pressed with the reset button, then reset button should be released firstly; after MCU circuitry starts to work, boot loader button also can be released too. During the power up of the circuitry, if boot loader button haven't been pressed, the board will work at application mode. At this mode, boot loader button could be used for general usage.

The application firmware is written as communication device class (CDC). By this class emulation of virtual serial port is done. Interrupt type USB communication is used.

The initialization of variables and registers are done at the start of the firmware. Then, a main loop starts to execute regular checks for USB service. If there is data at the buffer its first byte is tested if it is a predefined initial character for start of execution. After getting start byte, on board output resources such as leds, analog

digital outputs are updated due to commands came from USB. Then, data transfer routine via SPI is executed as USB Board is SPI master. SPI master checks whether there is a connected peripheral exist. There is timeout span for this check. If timeout occurs, master checks other peripheral connectors or continue other instructions.

Flowchart of firmware is presented in Figure 3.6. From this flowchart it could be seen that for SPI communication routine, only one peripheral communication is shown. This routine repeats itself 4 times at 4 peripheral connectors. Data transfer is done according to a protocol of which will be explained later. If there is/are a peripheral/s on the bus, its information is written to USB out buffer. Then on board input resources such as button, pot, analog digital inputs are written to USB out buffer. At the end, USB out buffer is sent to PC via USB and returns to main loop for data reception.

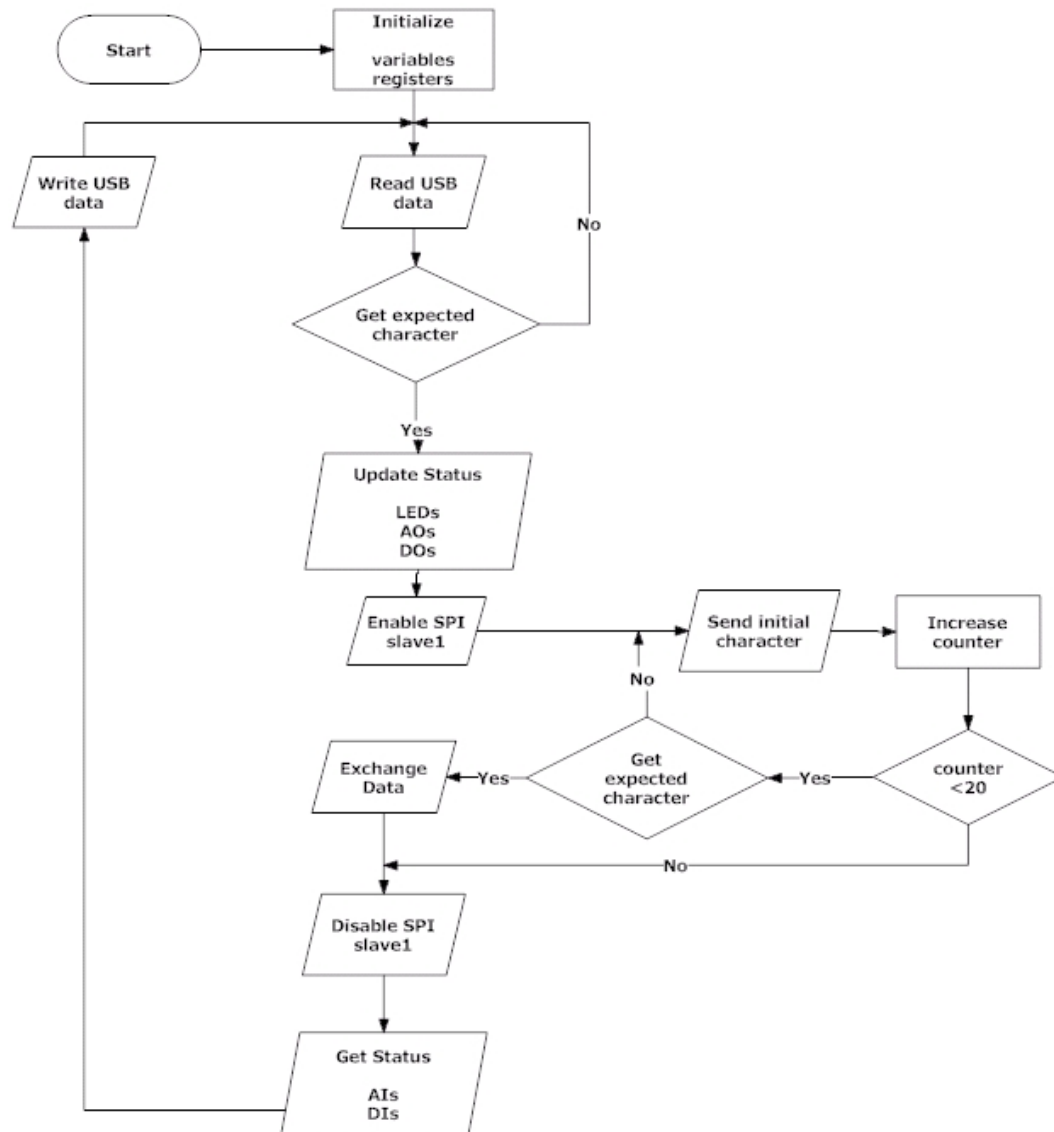


Figure 3.6 Flow diagram of USB Board firmware

Data transfer between USB Boards and PC is done via a protocol. There are two protocol descriptions according to data is received or sent. The details of protocol are presented in table 3.2. Initially, a starter character is sent from PC to USB Board and then data related to USB Board resources is sent. After that, the peripheral boards' data is sent. This is nearly the same sequence for data transfer protocol for data transfer from USB Board to PC.

Table 3.3 USB Board - PC communication protocol

USB Board data transfer			
Bytes	Received Data	Bytes	Sent Data
0	starter	0 - 1	initials
1	Leds	2	button
2 – 3	PWMs	3	pot
4	Digital Outputs	4 - 8	ADCs
5 – 14	Device1 Data	9 - 18	Device1 Data
15 – 24	Device2 Data	19 - 28	Device2 Data
25 – 34	Device3 Data	29 - 38	Device3 Data
35 – 44	Device4 Data	39 - 48	Device4 Data
45 – 59	Unused	49 - 59	Unused

There are four connections for SPI communication on the USB Board. According to boards' resources, data transfer bandwidth is determined as 10 bytes which is sufficient for their tasks. These tasks will be presented at following chapters. There are also unused additional bytes at the protocol for future development and expansions.

3.3 Conclusion

For PC interfacing of the designed peripheral systems a board with USB communication capability is designed. First version of this board (Figure 3.1) is also presented for the users who want to use it for simple applications or with breadboard usage because of its easy interfacing to breadboard.

USB Board has six identical ribbon cable connectors for peripheral boards and other external interactions. And there is a SMPS unit supplying power 5V capable of up to 3A loads for peripheral boards and/or external usages. There is also a button, a pot and a tricolor led for simple on board applications.

Peripheral board communication is done by SPI. USB Board is SPI master and by the configuration of SPI peripheral boards can be connected to any four peripheral connectors with hot plug fashion. PC and peripheral boards' communications are done by a defined protocol.

At next chapters peripherals boards will be introduced. More information about USB Board can be found at appendices.

CHAPTER 4

ANALOG INPUT DIGITAL INPUT OUTPUT BOARD

Several peripheral boards are designed which are complementary for USB Board and also can be used as stand alone usage. The AIDIO Board is one of them and AIDIO is abbreviation of analog input digital input output. As the name implies that, this board is responsible for IO management of analog digital sources. At this chapter this board will be presented in this. The more specific information like design details are given at appendices.

4.1 Hardware

At this board, Microchip's 16F877 MCU is used. The reason of this selection is chip's popularity and availability at the market. With 40 pins its resources are sufficient for general IO applications.

There are four leds on the board for general usages. There are two seven segments for display purposes. These seven segments are driven by two serially connected shift register integrated circuits (74HC595) by three pins at the MCU (Figure 4.1). Serially bytes are shifted at shift registers and these bytes, equivalent to eight bit output, are sent to seven segment display units' pins via resistors. The other method of driving seven segments is using 14 pins from MCU. By using the implemented method, fewer pins are used and also the shift register concept can be covered as educational purposes.

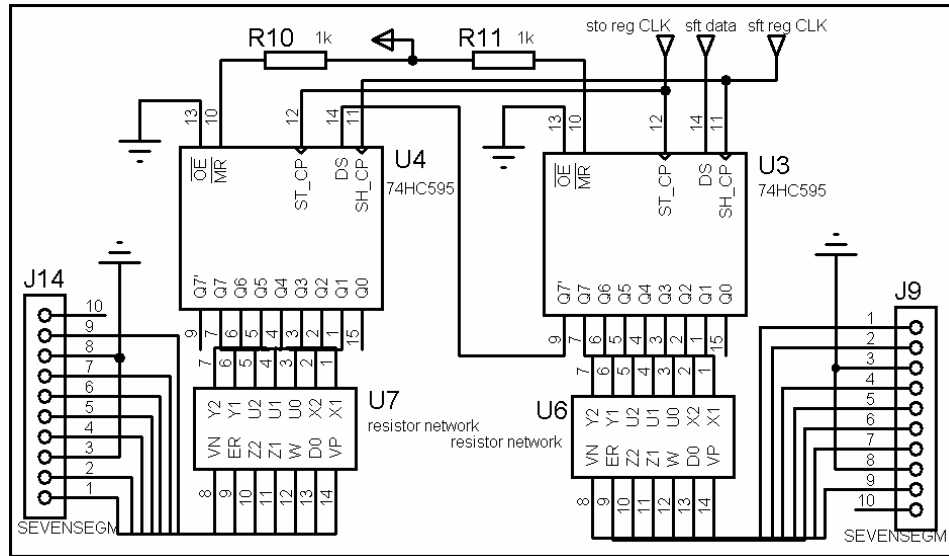


Figure 4.1 Seven segment display driver

For driving high voltage or high current devices like relays, RC servo motors, a Darlington transistor array integrated circuit ULN2003 is added on the board. For usages, four 12V double throw relays are put on the board. The remaining three outputs of ULN2003 are connected to female pin headers with power and ground supplies. By this way, RC servos can be connected to these connectors easily. The output driving power supply of this unit is separated from the power supply of the board. So to work with this unit, another power supply is to be connected to proper connections. And the two inputs of this unit are connected to MCU via jumpers, because of these two pins of MCU can be also as timer output. So depending on purposes these jumpers should be used.

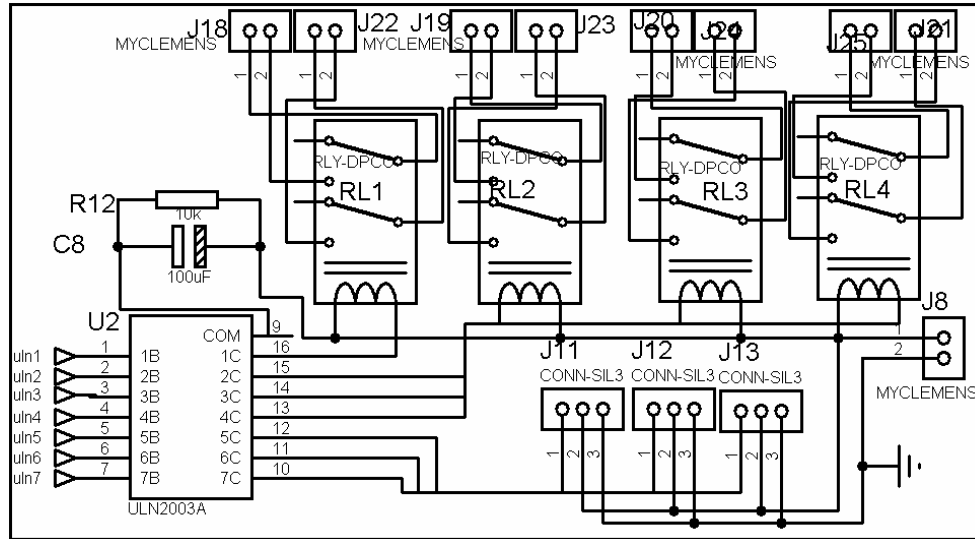


Figure 4.2 ULN2003 connection details

As digital input devices there are four push buttons which are parallel to four dual in-line package (dip) switches. To use push buttons, dip switches should be in open position. There are also two pots for analog reading which are connected to reference inputs of analog ports. So they can be used as analog inputs and/or to determine the limit voltages of other analog inputs. The remaining six analog ports are connected to female pin headers for external usage.

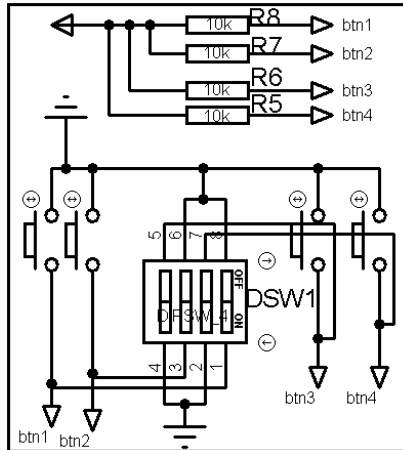


Figure 4.3 Parallel push buttons and dip switch

For power supply, a power selector jumper is added. The board can be powered via peripheral ribbon cable connector which its source is SMPS unit at USB Board; or can be powered via linear voltage regulator unit 7805. By 7805 board can be used as stand alone. Also there is a power status led on the board. A reset push button is present for resetting the MCU.

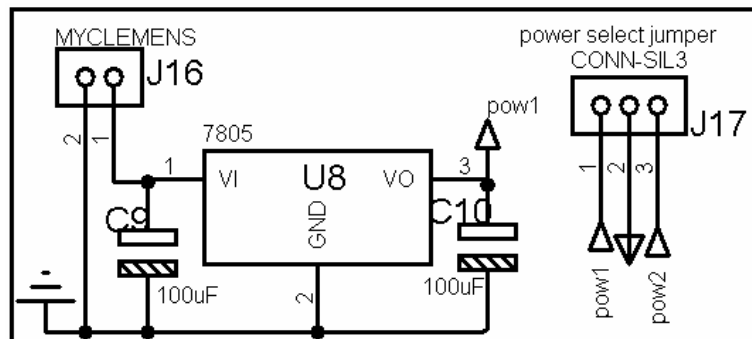


Figure 4.4 Linear voltage regulator

There is a RS-232 communication unit on the board. Its usage is foreseen for stand alone mode. But it can also be used for integrated usage with USB Board with a little change on the firmware.

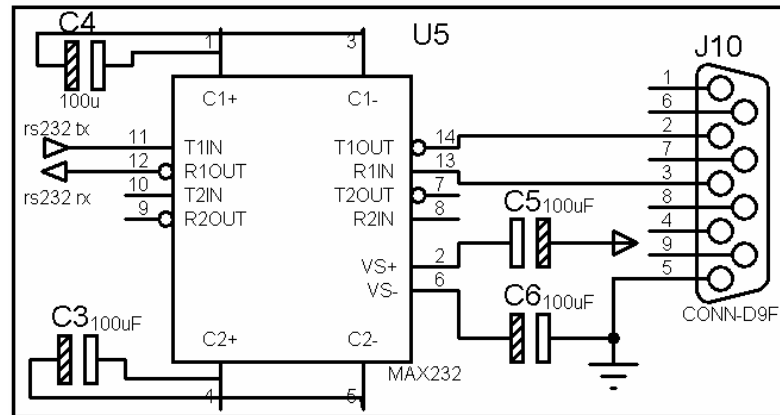


Figure 4.5 RS-232 communication unit

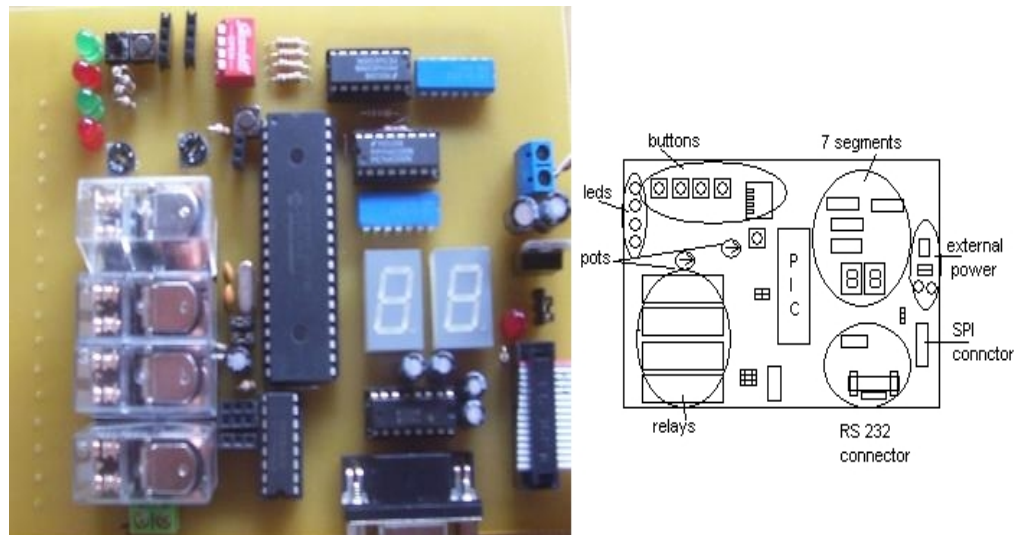


Figure 4.6 AIDIO Board and its layout

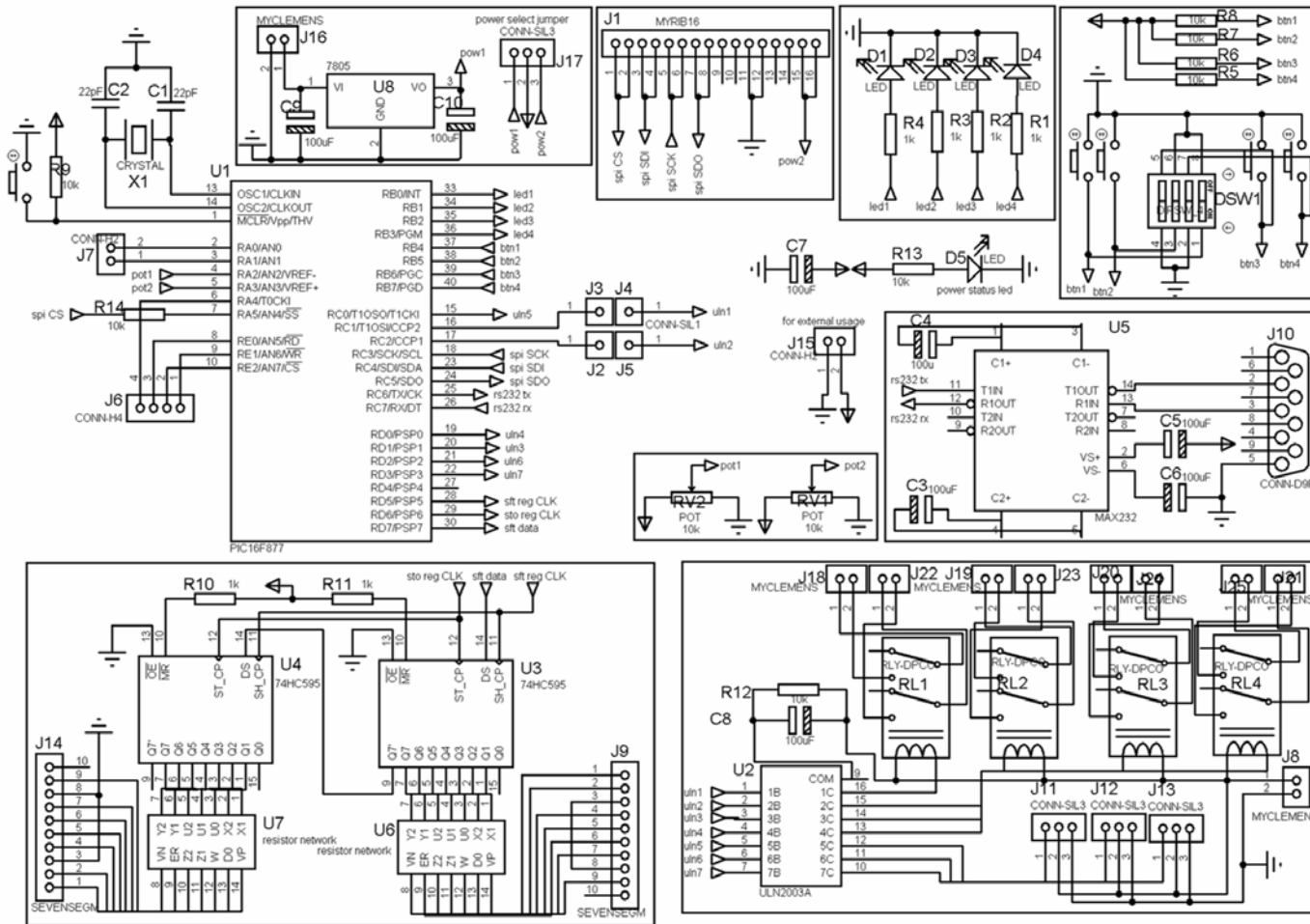


Figure 4.7 Schematic of AIDIO Board

Table 4.1 PIC 16F877 pin usages at AIDIO Board

PIC 16F877 pins usage					
description	connected to	pin #	pin #	connected to	description
resetting system	reset push button	1	40	button 1 and dip switch	general usage
general usage	external usage pin	2	39	button 2 and dip switch	general usage
general usage	external usage pin	3	38	button 3 and dip switch	general usage
general usage & Vref -	pot1	4	37	button 4 and dip switch	general usage
general usage & Vref +	pot2	5	36	led 1	general usage
general usage	external usage pin	6	35	led 2	general usage
spi SS	peripheral ribbon cable connector	7	34	led 3	general usage
general usage	external usage pin	8	33	led 4	general usage
general usage	external usage pin	9	32	supply voltage	
general usage	external usage pin	10	31	ground	
	supply voltage	11	30	74HC595 input	shift register data
	ground	12	29	74HC595 input	store register clock
	oscillator	13	28	74HC595 input	shift register clock
	oscillator	14	27	unused	
general usage	ULN2003 input	15	26	MAX232	RX
general usage	ULN2003 input via jumper	16	25	MAX232	TX
general usage	ULN2003 input via jumper	17	24	peripheral ribbon cable connector	spi SDO
spi SCK	peripheral ribbon cable connector	18	23	peripheral ribbon cable connector	spi SDI
general usage	ULN2003 input	19	22	ULN2003 input	general usage
general usage	ULN2003 input	20	21	ULN2003 input	general usage

A ribbon cable connector is put on the board for USB Board communication via SPI and power transmission. This connector details are given at USB Board section.

4.2 Firmware

The firmware starts with initialization of variables and registers of MCU. Then main loop is started with fetching of statuses of inputs like pots, buttons. This collected data will be sent to SPI routine. At SPI routine, SPI buffer is checked by polling method. If starter byte is received data transfer is executed. During this process, 10 byte of data is exchanged. This 10 byte description is presented at Table 4.2.

Table 4.2 AIDIO Board – USB Board communication protocol

AIDIO Board data transfer			
Bytes	Received Data	Bytes	Sent Data
0	starter	0	initials
1	leds + relays	1	ADCs
2	7 segment 1	2	ADCs
3	7 segment 2	3	pot1
4	Unused	4	pot2
5	Unused	5	buttons
6	Unused	6	Unused
7	Unused	7	Unused
8	Unused	8	Unused
9	Unused	9	Unused

As it can be seen from protocol table, some bytes are unused so that there is room for future developments. After SPI communication routine status of outputs like, leds relays are updated due to command come from USB Board. Then execution returns to the main loop.

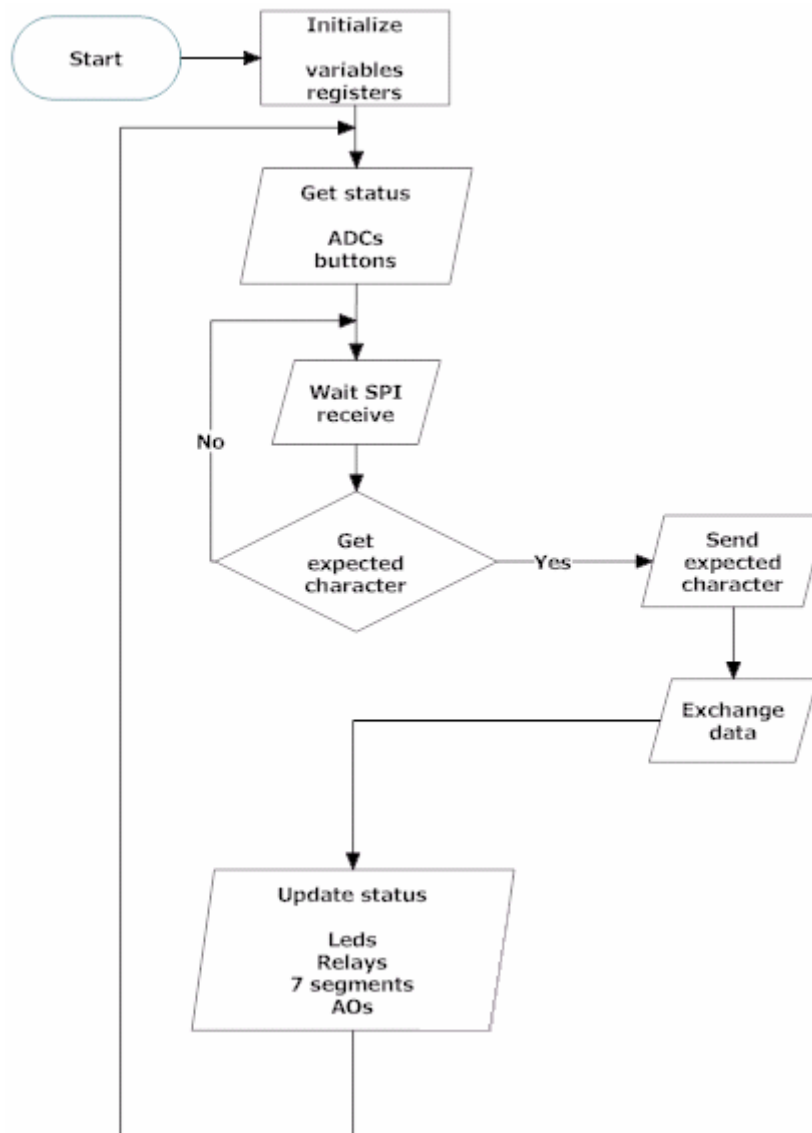


Figure 4.8 Flowchart of AIDIO Board firmware

4.3 Conclusion

AIDIO Board is one of the peripheral boards can be used for general IO applications. The Board can be connected to USB Board for PC communication and for power needs or can be used stand alone with on board power regulator and RS-232 communication unit.

On the board there are leds, buttons, seven segments, relays, pots which are ready for use. For USB Board connection there is one ribbon cable connector which it can be connected to any proper connectors of USB Board. Communication with USB Board is done by SPI via a defined protocol and AIDIO Board is one of the SPI slaves on the over all system.

CHAPTER 5

MOTOR BOARD

Another designed board for peripheral usage is Motor Board. This board is responsible for motor control applications. In this chapter, some information about board's hardware and firmware will be presented. Detail information about board can be found in the appendices.

5.1 Hardware

The MCU of this board is Microchip's 16 bit digital signal controller 30F4011. This chip's some hardware features like 6 PWM channel with complementary or independent output modes and quadrature encoder interface with phase A, B and index inputs; make this MCU ideal for motor control applications.

For dc motor driving purposes a dual full bridge driver L298 is added to the board. This driver is capable of driving two motors. This driver operating voltage is 46V with maximum 4A total current. Its supply voltage is separated from MCU circuitry and a connector is present for external power supply connection.

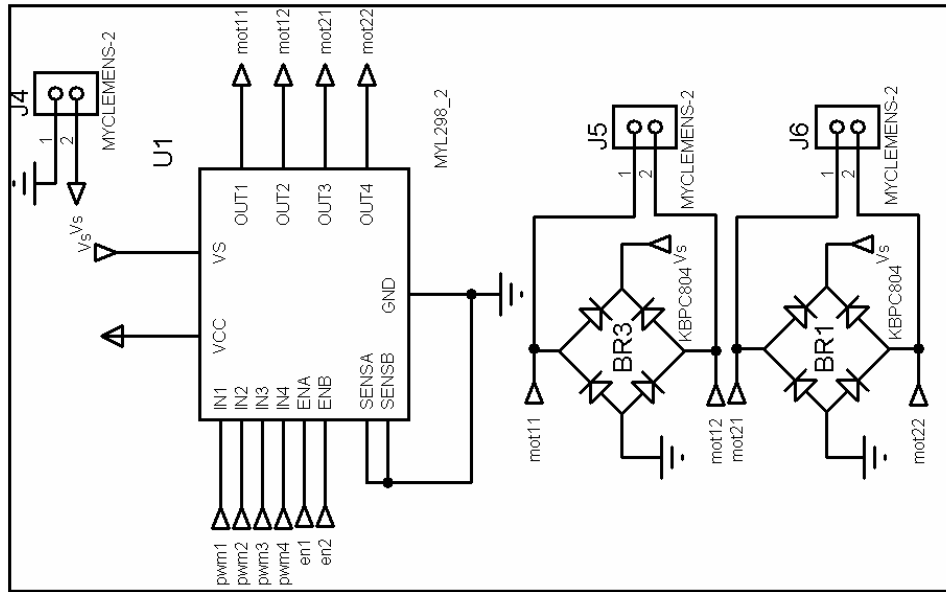


Figure 5.1 Motor driver unit

Two encoder interfaces is added on the board. One is directly connected 30F4011's quadrature encoder interface pins. The other is connected to interrupt pins via a D flip-flop. The first interface enables the use of three inputs (phase A, B and index) usage, the second one is for two inputs (phase A and B). For power needs of encoders also power select jumpers with external power connection connectors is added.

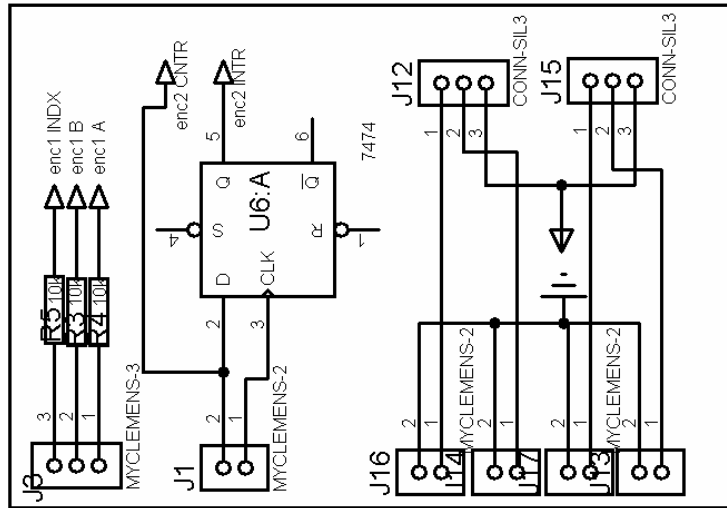


Figure 5.2 Encoders interface

There is also a digital analog converter (DAC) unit on the board which is Microchip's MCP4922. It has two DAC channels with 12 bit resolution and it communicates via SPI. Connectors are put on the board for these output channels interfacing. This unit usage is for motor control applications to supply inputs of other motor driver circuits.

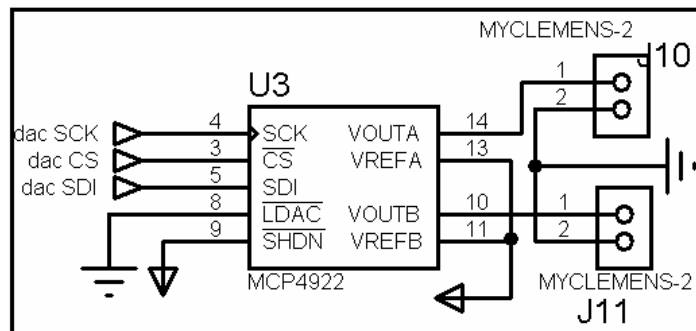


Figure 5.3 DAC unit

For stand alone usage option, there is a linear voltage regulator 7805 on the board which is selected by jumper. The other power supply is peripheral connector cable comes from SMPS unit on the USB Board. A reset button is also present for resetting the board.

Like at the AIDIO Board, there is a unit for RS-232 communication that can be used at stand alone usage or integrated usage with USB Board by a little change on the firmware.

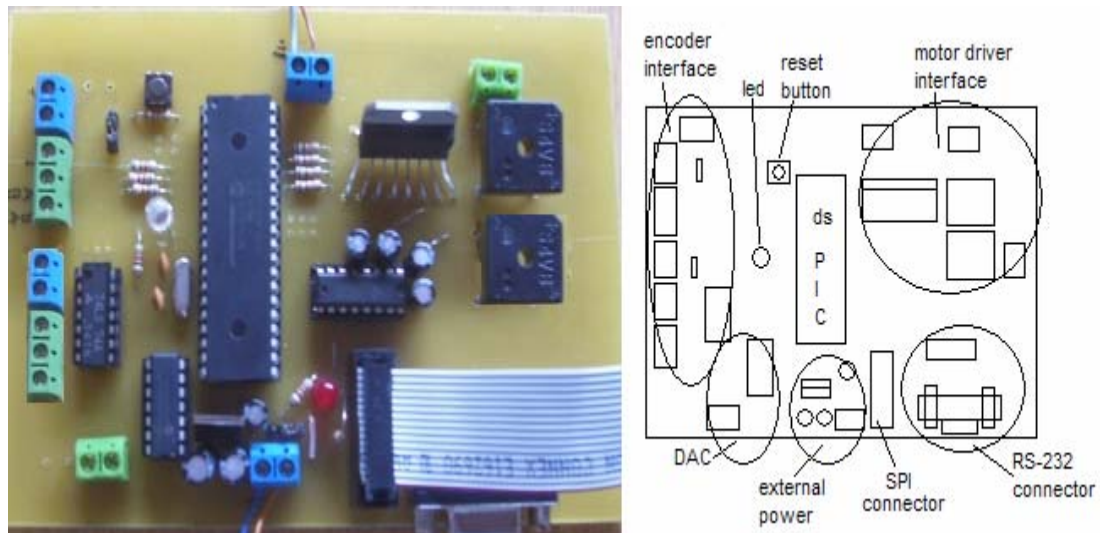


Figure 5.4 Motor Board and its layout

Figure 5.5 Schematic of Motor Board

Table 5.1 PIC 30F4011 pin usages at Motor Board

PIC 30F4011 pins usage					
description	connected to	pin #	pin #	connected to	description
resetting system	reset push button	1	40	supply voltage	
general usage	external usage pin	2	39	ground	
general usage	external usage pin	3	38	L298 pwm input	PWM1
spi SS	peripheral ribbon cable connector	4	37	L298 pwm input	PWM2
QEI index	encoder1 connector	5	36	L298 pwm input	PWM3
QEI A	encoder1 connector	6	35	L298 pwm input	PWM4
QEI B	encoder1 connector	7	34	external usage pin	PWM5
general display usage	tri-colored led	8	33	external usage pin	PWM6
general display usage	tri-colored led	9	32	supply voltage	
general display usage	tri-colored led	10	31	ground	
	supply voltage	11	30	L298 enable input	enable1
	ground	12	29	L298 enable input	enable2
	oscillator	13	28	MAX232	RX
	oscillator	14	27	MAX232	TX
	unused	15	26	peripheral ribbon cable connector	spi SDI
phase A	encoder2 connector	16	25	peripheral ribbon cable connector	spi SDO
phase B	encoder2 connector	17	24	peripheral ribbon cable connector	spi SCK
DAC spi SCK	MCP4922	18	23	unused	
DAC spi SDI	MCP4922	19	22	MCP4922	DAC spi SS
	ground	20	21	supply voltage	

5.2 Firmware

Similar to the other boards' firmware, this firmware starts with initialization of variables and registers. The overall algorithm is same with AIDIO Board. After initialization main loop starts. The sequence at the main loop is as follows: getting on board inputs (encoders), SPI communication routine and at last updating on board outputs (leds, motors, DAC) then returning to the start of loop.

The SPI routine is also same. When SPI buffer is full, its value is checked. If the expected character is received, 10 bytes of data is exchanged. The details of this transfer are shown at table 5.2.

Table 5.2 Motor Board – USB Board communication protocol

Motor Board data transfer			
Bytes	Received Data	Bytes	Sent Data
0	Starter	0	Initials
1	Leds	1	Encoder1
2	Motor enables directions	2	Encoder1
3	PWM 1 value	3	Encoder2
4	PWM 1 value	4	Encoder2
5	PWM 2 value	5	Unused
6	PWM 2 value	6	Unused
7	DAC value	7	Unused
8	DAC value	8	Unused
9	Unused	9	Unused

At the sent data protocol first byte is reserved for board detection byte like the other boards. And in the protocol, some bytes are reserved that may be used for future developments.

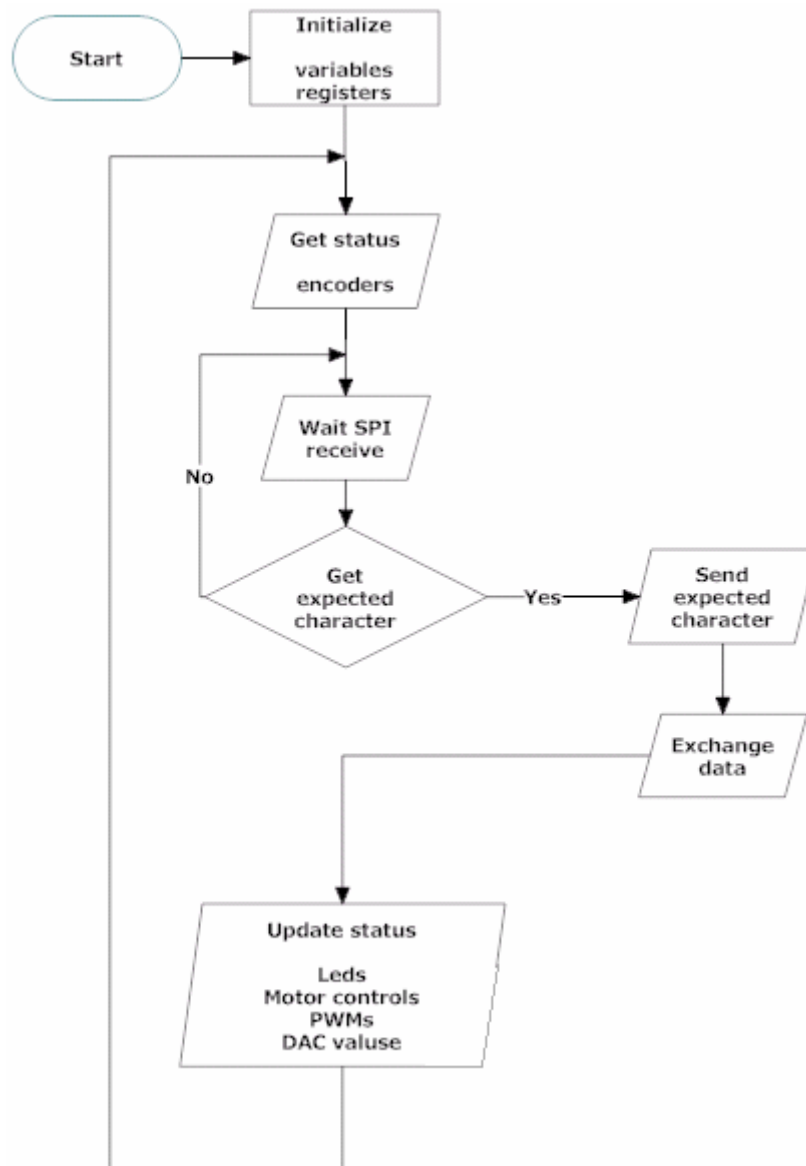


Figure 5.6 Flowchart of Motor Board firmware

5.3 Conclusion

Motor Board is designed for controlling DC motor, bipolar step motor or even for RC servo motors. Microchip's 16 bit 30F4011 dsPic family chip is used as MCU of this board. Its dedicated PWM outputs and QEI input with 48 Kb on chip flash program space, 2 Kb on chip RAM make this MCU ideal for motor control applications. In addition to its PWM and QEI ports, a DAC chip and a D type flip flop is added for motor control and encoder interfacing.

Like other boards it can be used stand alone by on board power regulator and RS-232 connector. At default configuration Motor Board is connected to USB Board by ribbon cable which carries data and MCU power.

Motor Board communication with USB Board is done by SPI. Motor Board is one of the SPI slaves in the system and like other peripheral boards it can be connected any proper connectors on the USB Board.

CHAPTER 6

LCD BOARD

The last designed board is LCD Board. This board is designed to be used for display purposes of the other boards. In this chapter, information related to this board's hardware and firmware will be presented. Detail information can be found in appendices.

6.1 Hardware

This board usage is designed for driving dot matrix LCD with 6 pin interface. As a result, the demands from MCU are having 6 digital ports and able to communicate via SPI. So Microchip's 16F88 product with 18 pin is selected which is an enhanced version of famous product PIC 16F84.

A linear voltage regulator 7805 is added for stand alone usage on the board. Like the other peripherals a power supply select jumper is present. Peripheral ribbon cable connector is also carries power from SMPS unit at USB Board. There is also a reset push button for resetting the board.

For LCD connection, a 14 female pin header is attached. Also a pot is added to configure the contrast of LCD.

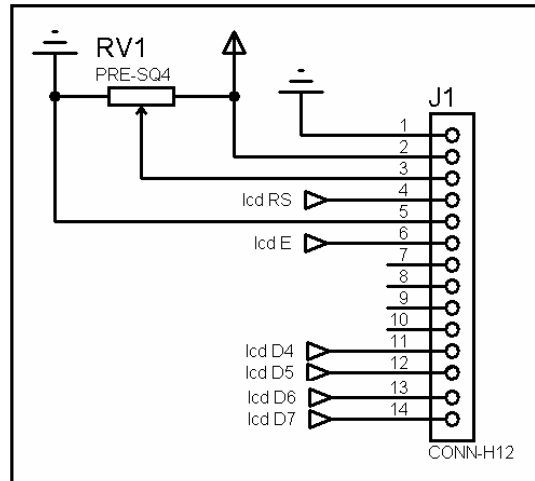


Figure 6.1 LCD connection details

The 3 unused pins are connected to pin header connectors. These pins can be used for RS-232 communication if it is needed. Besides, there is a ground connector for external usage

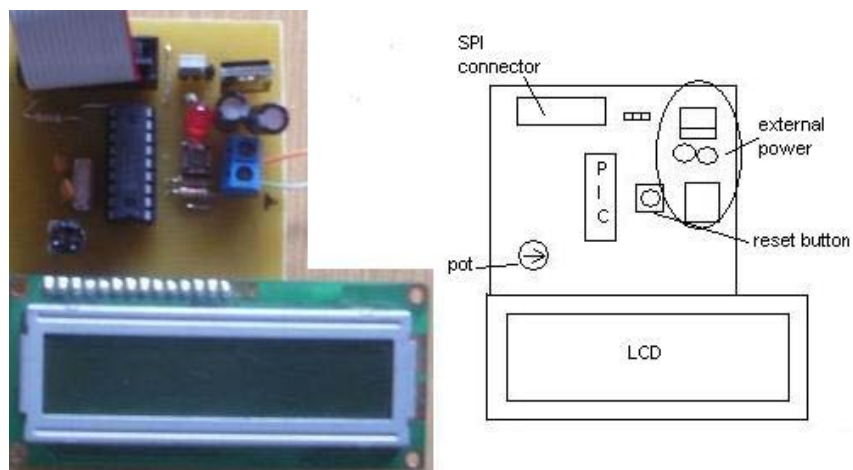


Figure 6.1 LCD Board and its schema

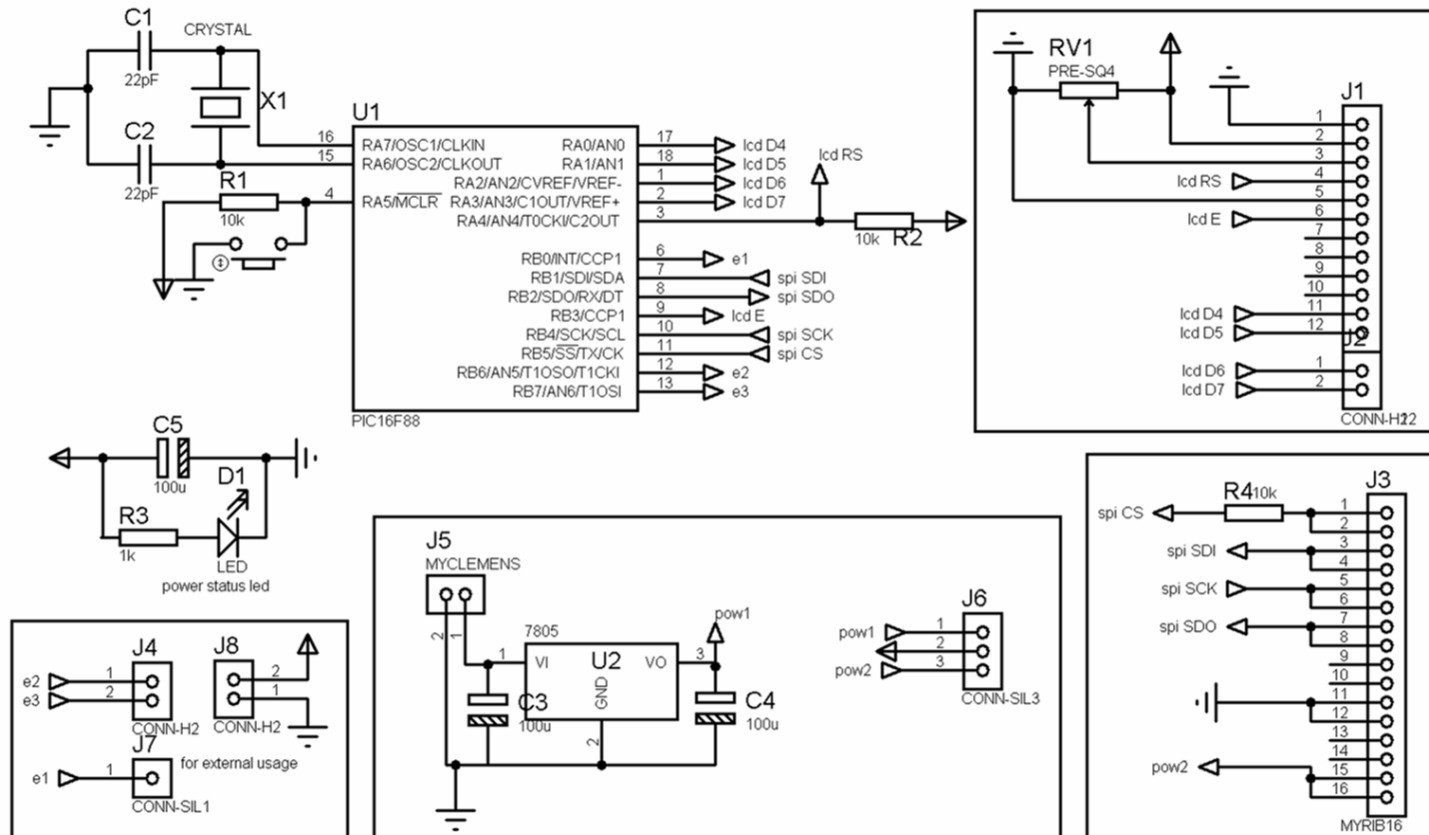


Figure 6.3 Schematic of LCD Board

Table 6.1 PIC 16F88 pin usages at LCD Board

PIC16F88 pins usage					
description	connected to	pin #	pin #	connected to	description
D6	LCD	1	18	LCD	D5
D7	LCD	2	17	LCD	D4
RS	LCD	3	16	oscillator	
resetting system	reset push button	4	15	oscillator	
	ground	5	14	supply voltage	
general usage	external usage pin	6	13	external usage pin	general usage
spi SDI	peripheral ribbon cable connector	7	12	external usage pin	general usage
spi SDO	peripheral ribbon cable connector	8	11	peripheral ribbon cable connector	spi SS
E	LCD	9	10	peripheral ribbon cable connector	spi SCK

6.2 Firmware

The firmware is similar to the other peripherals board, except this board doesn't have any input data to transfer to the USB Board. It takes 10 bytes from SPI bus and processes this data and updates the status of LCD and sends the initials of LCD Board for board detection.

Before explaining the updating mechanism of LCD, brief information about LCD displays will be given for better understanding of the process. LCDs are controlled via its registers and instructions supplied by the MCU of the LCD. Communication interfaces are parallel bus with 8 bit or 4 bit length. If 4 bit mode is used, the data is sent at two cycles.

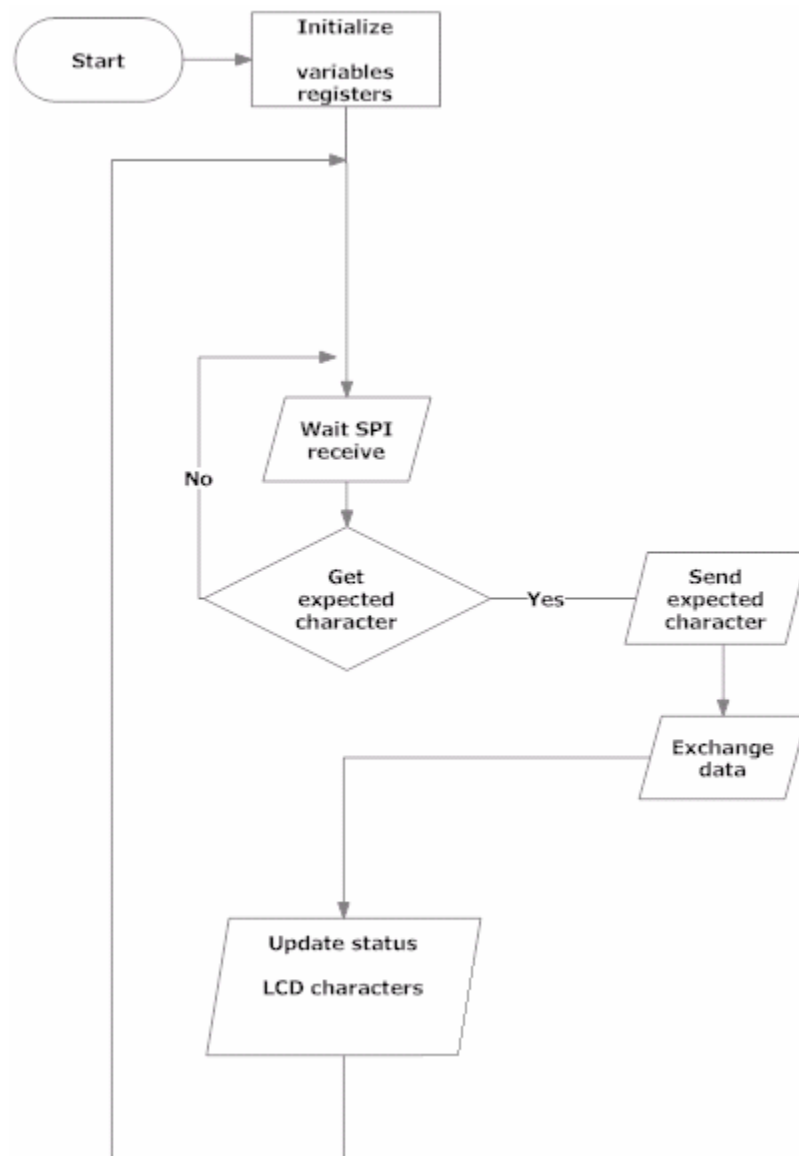


Figure 6.4 Flowchart of LCD Board firmware

There are several instructions like clearing display, blinking cursor, setting cursor moving direction, showing character on display etc. So for LCD controlling three type of usage is designed.

The first byte of SPI data determines the mode of usage. At first mode display is cleared and the remaining nine bytes of SPI data is displayed on the LCD. So at this mode only first nine characters on the LCD is used. At second mode character display is done via addressing via two byte series. First byte is the address of the LCD character to be displayed at. The second one is character value. Remaining eight byte is used at this mode, displays four character at any desired location on the LCD. The last mode is instruction mode. At this mode eight instructions is sent to LCD. These instructions are: clear display, return home, cursor off, underline cursor on, blinking cursor on, move cursor left one position, move cursor right one position, move cursor to the beginning of second line.

Table 6.2 LCD Board – USB Board communication protocol

bytes	1st mode	2nd mode	3rd mode
0	mode select byte		
1	char value	address	command
2	char value	char value	unused
3	char value	address	unused
4	char value	char value	unused
5	char value	address	unused
6	char value	char value	unused
7	char value	address	unused
8	char value	char value	unused
9	char value	unused	unused

6.3 Conclusion

LCD Board is designed for display usages. For this purposes most common type 14 pin LCD is used. For the moment 16x2 LCD display is used with the board. LCD Board's MCU's unused pins are connected to connectors for future additional implementations.

Like other peripheral boards LCD board is communicate with USB Board by SPI and can be used stand alone. Unlike the other peripheral boards LCD Board is responsible only for one purpose which is LCD control. This little board is also a good example for designing a dedicated board for this thesis system.

CHAPTER 7

PC INTERFACING AND SOFTWARE DESIGN

This thesis's start point is to develop a system that covers the mechatronic education in a hybrid paradigm (mixture of low and high level programming). As a result of this approach, at PC side a dynamic link library (DLL) is written. So users can access to boards' resources easily via this DLL. The DLL is written at .NET framework C#. But it can be used at any .NET framework language. Also to demonstrate the usage of the DLL a sample application program is written.

Before giving details of the DLL, some other information should be explained about USB communication. As it mentioned before, two types of firmware (boot loader and application) are present at USB Board's MCU. So there are two types of drivers should be used. For USB communication Microchips' drivers are used. Although advantages of using a custom USB driver, developing a custom USB driver is foreseen as time taking process in the content of this thesis. For boot loader program, Microchip's supplied graphical user interface (GUI) is used. The driver of application program is Communication Device Class (CDC) driver of Microchip. This CDC driver with the firmware emulates the serial port on PC and creates virtual serial port when MCU is plugged to PC. So users past projects about serial port usage can be used with minor modifications. For application program a DLL is written for users to write their GUIs with examples of its usage. The overall work of thesis at PC side is shown at Table 7.1.

Table 7.1 PC side software development sources

	driver	DLL	GUI
boot loader	Microchip	Microchip	Microchip
application	Microchip	thesis	thesis/users

Another issue is multiple usage of USB Board. Each USB device has to have a unique identifier (ID). This identifier is consisting of vendor ID and product ID. To get a vendor ID one should give at least \$2000 to USB Implementers Forum, Inc., creators of USB technology. But Microchip is supplying its vendor and product IDs. So to connect multiple USB Boards to PC product IDs should be changed. To change product ID, firmware of MCU and the INF extension files of drivers should be updated. More details can be found at appendices.

7.1 DLL Methods

At DLL there are fields and methods (Figure 7.1). All fields are set as private so they can't be reached from users. These fields are consisting of serial port object, configuration data of serial port, buffers of serial port. Serial port and buffers are made accessible through DLL. So all methods in the DLL are working on the same serial port and serial port data. The methods are classified according to usage and presented as with these classifications. All the methods are listed at Table 7.2.

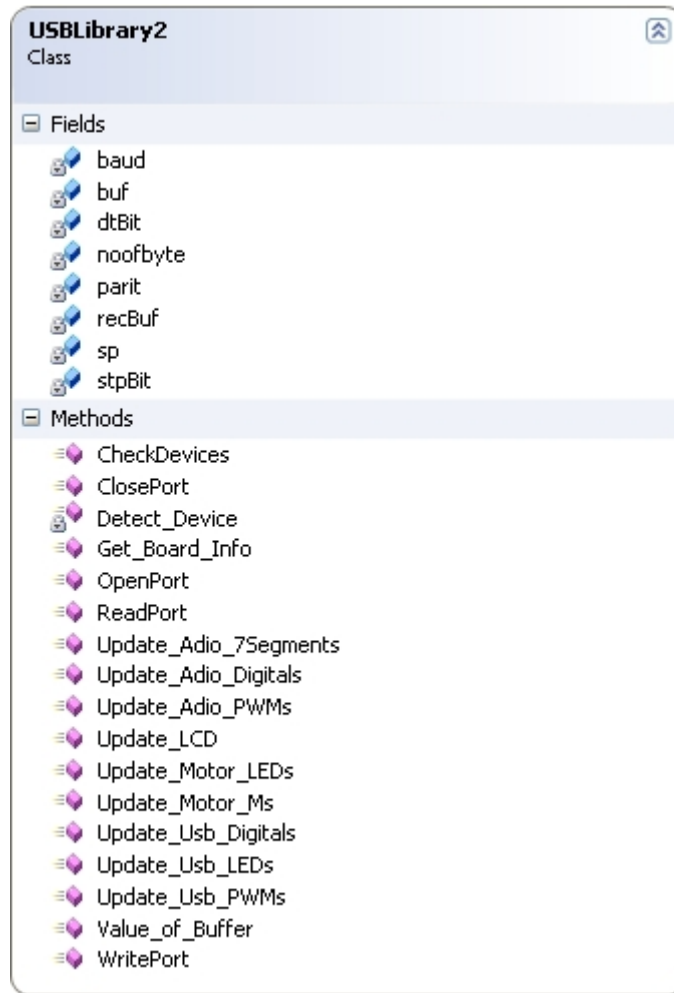


Figure 7.1 Class Diagram of the DLL

Table 7.2 DLL Methods

return type	name	arguments
string[,]	CheckDevices	void
string	Value_of_Buffer	void
string	OpenPort	string portno
string	ClosePort	string portno
void	WritePort	string data
string	ReadPort	void
void	Update_Usb_LEDs	bool led1, bool led2, bool led3
void	Update_Usb_PWMs	int pwmvalue1, int pwmvalue2
void	Update_Usb_Digitals	bool DO1, bool DO2, bool DO3, bool DO4, bool DO5, bool DO6, bool DO7, bool DO8
void	Update_LCD	int device, string lcdText
void	Update_Audio_Digitals	int device, bool uln1, bool uln2, bool uln3, bool uln4, bool led1, bool led2, bool led3, bool led4
void	Update_Audio_7Segments	int device, int seg1, int seg2
void	Update_Audio_PWMs	int device, int pwm1, int pwm2, int pwm3
void	Update_Motor_LEDs	int device, bool led1, bool led2, bool led3
void	Update_Motor_Ms	int device, bool en1, bool dir1, bool en2, bool dir2, int pwmval1, int pwmval2
string[]	Get_Board_Info	void
string	Detect_Device	int sector , char initial

7.1.1 Debug Related Methods

CheckDevices: Searches the plug and play devices and filters them according to their device IDs to find the USB devices with a specified vendor ID. And return the result as string array. If device(s) is/are found method sent the serial port address of device which is needed later for communication and its device ID for debugging.

Value_of_Buffer: Returns the serial port sent buffer values as a string. Used for methods affects on data buffer.

7.1.2 Serial Port Related Methods

Open_Port: It takes serial port address of which is to be opened. Usage with 'CheckDevices' method is recommended. First a simple check is done for proper port address is received then that port is configured at predefined values and tried to be opened. Results are returned.

Close_Port: Similar to 'Open_Port' method except this one tries to close the port.

Write_Port: Sets the first value of sent buffer according to protocol then write the sent buffer which is modified at "Updating Status of Boards Related Methods" to serial port sent buffer.

Read_port: Reads the serial port receive buffer and returns it value as string

7.1.3 Updating Status of Boards Related Methods

On USB Board there are four connectors for peripheral boards. So at peripheral board related methods there is a need for addressing. Although all peripheral can be identified via the protocol, when 4 one type of board (e.g. 4 ADIO Boards) connect to these four peripheral connectors there is a need for addressing.

7.1.3.1 USB Board Related Methods

Update_USB_LEDs: There is one tri colored led on USB Board. This method updates the status of this led. Takes three boolean inputs and sets bits of related byte in the protocol.

Update_USB_PWMs: Sets the two hardware PWM pin of USB Board value. Takes two integer value converts them to byte equivalent and place it to the proper buffer address.

Update_USB_Digitals: Takes eight bits information in boolean form and sends this information as byte value to USB Board to update its unused digital output resources.

7.1.3.2 ADIO Board Related Methods

Update_ADIO_Digitals: First takes integer value between 1 and 4 which determines the peripheral device connected connector. This information used to put the related data to proper address at the protocol. Then takes eight bits information in boolean form. First four bit is about relay statuses, last four bit is about led statuses of ADIO Board. Then converts these bits to byte equivalent value.

Update_ADIO_PWMs: First takes integer value between 1 and 4 to determine connected peripheral connector. Then takes three bytes of information in integer form. These values are used to update PWM values of ADIO Board.

Update_ADIO_7Segments: First takes integer value between 1 and 4 to determine connected peripheral connector. Then takes two byte of information in integer form. These values are used to update the 2 seven segment display values on ADIO Board.

7.1.3.3 Motor Board Related Methods

Update_Motor_LEDs: First takes integer value between 1 and 4 to determine connected peripheral connector. There is one tri colored led on Motor Board. This

method updates the status of this led. Takes three boolean inputs and sets bits of related byte in the protocol.

Update_Motor_Ms: First takes integer value between 1 and 4 to determine connected peripheral connector. This method is used for motor driving of two motors at Motor Board. Takes four bits for motor 1 enable and direction, motor 2 enable and direction information. Then takes two integers ranging between 0 and 2000 to determine the PWM values of motors. These integer values are split into two byte value according to a formula to get the same value at MCU by calculating the inverse of the formula later.

7.1.3.4 LCD Board Related Methods

Update_LCD: First takes integer value between 1 and 4 to determine connected peripheral connector. This method takes string value and sends first 10 characters to the LCD Board.

7.1.4 Getting Status of Boards Related Methods

Get_Board_Info: This method is used for data reading of the boards. Returns string array which its length is five. These five strings have information of USB Board and four peripheral connectors. These strings are consisting of board type and its specific information like buttons, pots statuses etc. Data in these strings are separated by semicolons. For peripheral board detection 'Detect_Device' method is used.

Detect_Device: This method determines the peripheral boards connected to the connector of USB Board. This detection is done checking the initials of the boards sent data. After detection according to boards' resources, information strings are created mentioned at 'GET_Board_Info'. This method is a private method for users and used by 'GET_Board_Info' method.

7.2 Sample GUI

During the design stages of testing the hardware and software several GUIs are designed and used. One of them is introduced at this section as demonstration of DLL usage.

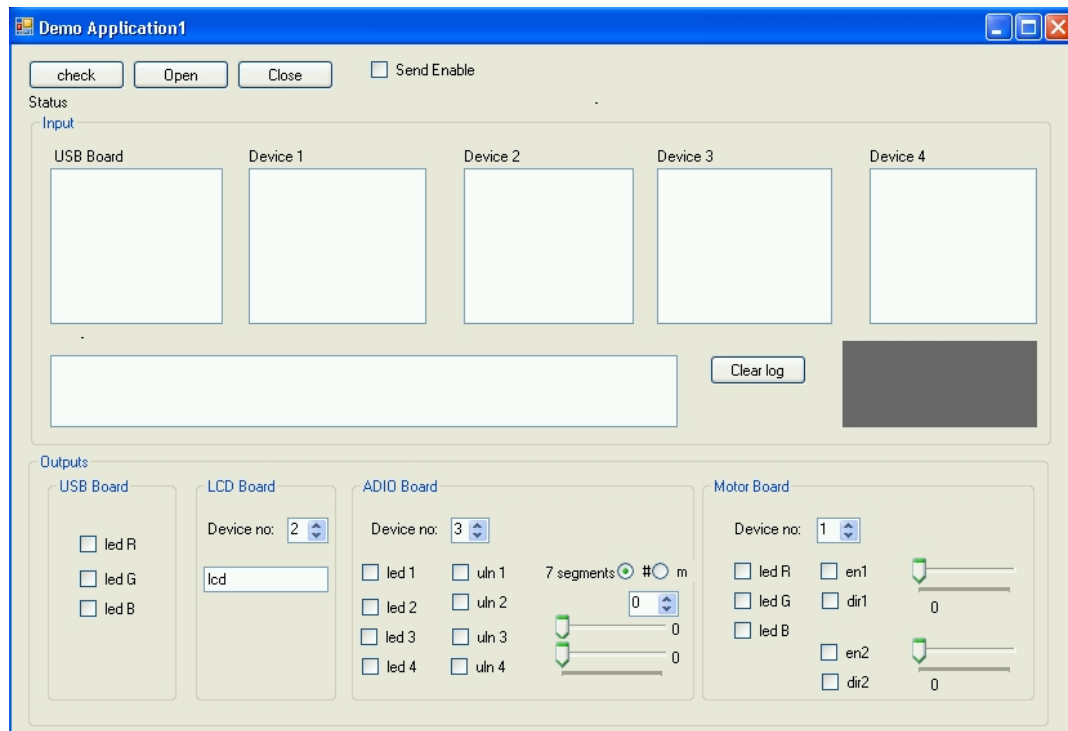


Figure 7.2 Sample GUI

The sample GUI is written at C# as a windows application. To use the DLL it has to be added to application references. At the start of the form class of the application the DLL object instance is created. At this point in the DLL is accessible throughout the form class.

There is a button to check the USB Boards is connected or not and buttons opens or closes the virtual port of connected USB Board. For statuses, information, warnings and errors there is a label under these buttons. After opening the virtual serial port to start the communication a check box is present.



Figure 7.3 Port related controls of GUI

When data transmission check box is checked the timer object which is created before is enabled. At timer object tick events; the virtual serial port receive buffer is read by proper DLL methods, displayed on GUI's control and port sent buffer is filled by GUI's controls and proper methods of DLL (polling method). USB Board and peripheral boards (that are connected to USB Board's four peripheral connectors) input resources are displayed on proper list boxes shown at Figure 7.4. Also virtual serial port receive buffer is displayed and logged for diagnostic purposes. There is a graphical control at GUI to display desired input change status in time.

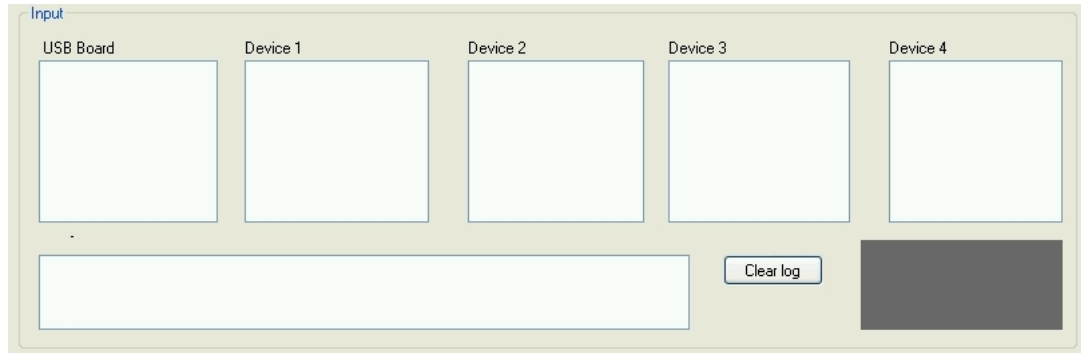


Figure 7.4 Input related controls of GUI

For output manipulation it is added controls related to boards. At this GUI, there is one control groups for one type of board. It is also possible that one type series of boards can connect to four peripheral connector of USB Board. So at this case same type control groups should be added to GUI. At each peripheral board control group there is a numeric up down box control for determining which device is connected to which connector of USB Board (Figure 7.5). And also there are several controls according to the type of the board.

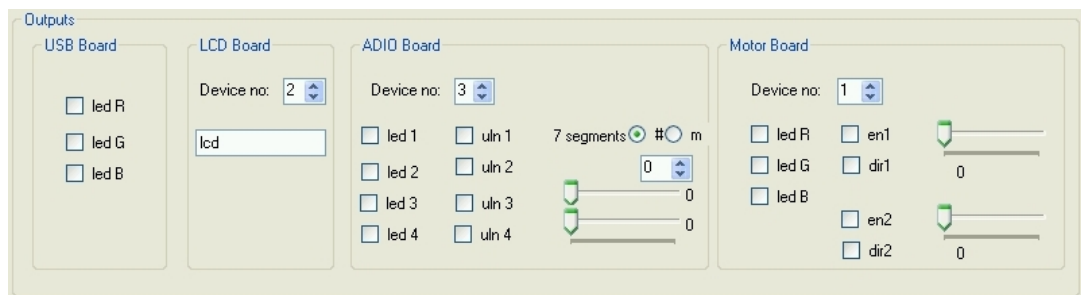


Figure 7.5 Output related controls of GUI

7.3 Conclusion

At the PC side design of the thesis a class library (a.k.a. DLL) is provided for end users. To explain the usage of this DLL with a hands on example, a sample GUI (based on the DLL) is implemented at C#. The reason of using C# is; it is one of the most commonly used object oriented language.

The DLL methods are written according to the protocol which is presented at the boards' chapters (chapters 3-6). Therefore, anyone who wants to develop projects on a PC by using these boards will need to know the proper use of DLL. However, if the user wants to write his/her custom library to meet some specific needs, it is also possible. At next chapter a demonstration of the overall system will be presented.

CHAPTER 8

CASE STUDY

To demonstrate the boards' capacity an experiment is designed. Also a lab sheet is prepared. At this experiment the boards gradually used one by one and at the end of the experiment all the boards are utilized together. By this way, smaller experiments can be derived, from one experiment.

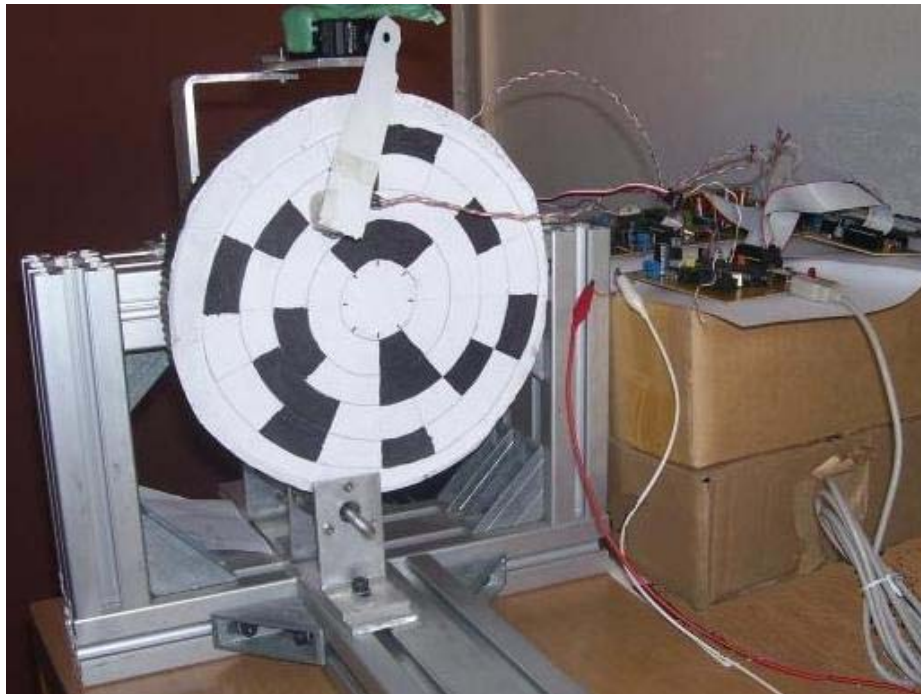


Figure 8.1 Experiment Setup

8.1 Experiment Description

The aim of the experiment is to simulate a hard disk reading. The hardware is a rotary disk driven by a 12V DC motor, an encoder, a switch, a reflective optical sensor mounted on the edge of a remote controlled (RC) servo motor and an encoded paper.

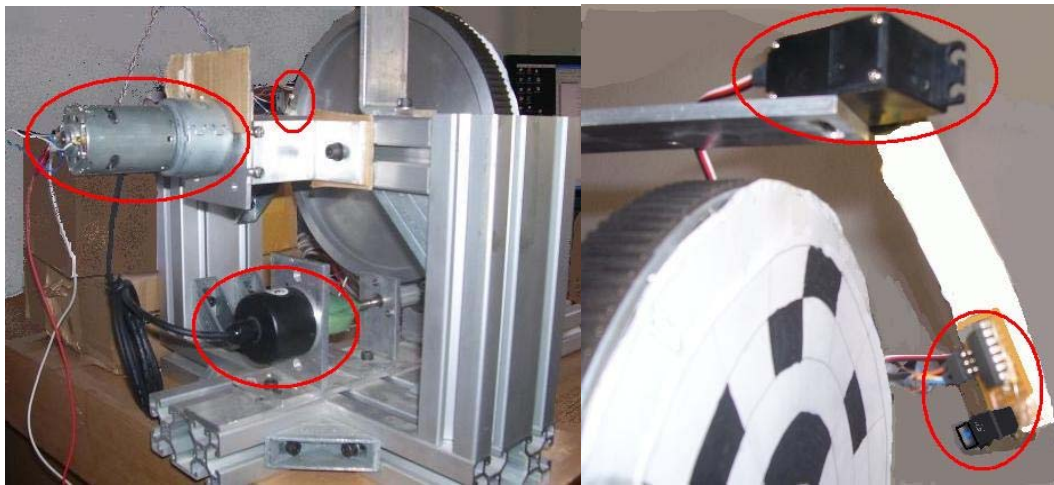


Figure 8.2 Sensors and actuators of the experiment setup

The hard disks are composed of sectors and tracks (Figure 4.3). For this experiment an encoded paper representing hard disk is prepared. Every region is representing one bit value. To read these bit values reflective optical sensors is used. The optical sensor is attached to a beam which is connected to the RC servo motor. By this way sensor could read any region on the disk. First servo motor is placed the optical sensor first track region and read all tracks at that circular region. This procedure is repeated at every track regions then from collected data

sector based information is achieved. For absolute positioning of sectors, a switch is present.

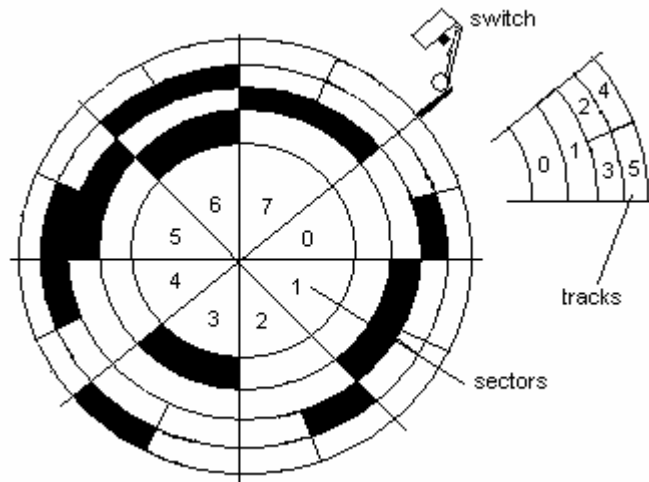


Figure 8.3 Encoded paper representing hard disk

8.2 Experiment Procedure

At the first stage of experiment reflective optical sensor and limit switch is interfaced to AIDIO Board. Then AIDIO Board is connected to the USB Board for PC communication. By rotating the disk manually observations on sensory reading are done.

At next section of the experiment, AIDIO Board seven segment displays are examined. Seven segment displays are consisting of seven leds and driven by shift register with a byte value. To displaying desired character at seven segment

display a lookup table is needed. So at this stage students are supposed to create this lookup table.

After creating lookup table, a counter display will be designed. The switch is generating signal at each turn of the disk. From this signal counting the result is supposed to be displayed at seven segment displays of the AIDIO Board. Rotating the disk manually counter display will be confirmed. The results also could be displayed at LCD Board.

To positioning the optical sensor by RC servo motor, the angular positions of the beam have to be determined. RC servo motor takes a byte value for setting angular position of its shaft. Like in the seven segment display numeric character determination, a look up table is needed to follow four circular regions.

Then the Motor Board is interfaced to the experiment setup. Encoder readings are observed by rotating the disk at desired PWM values via Motor Board. These encoder readings then sent to the LCD Board for display purposes. A simple controller could be implemented for speed control. Driving the disk at constant speed and positioning the RC servo motor to a proper position, the pattern on the disk is acquired by reflective optical sensor and a plot is drowned. Then this procedure is repeated for every track and sector based data is obtained.

8.3 Experiment Preparation and Results

To conduct the experiment necessary hardware connection is to be made. RC servo motor is connected to the USB Board. Switch and optical sensor is connected to AIDIO Board. DC motor and encoder is connected to the Motor Board. All peripheral boards are connected to the USB Board.

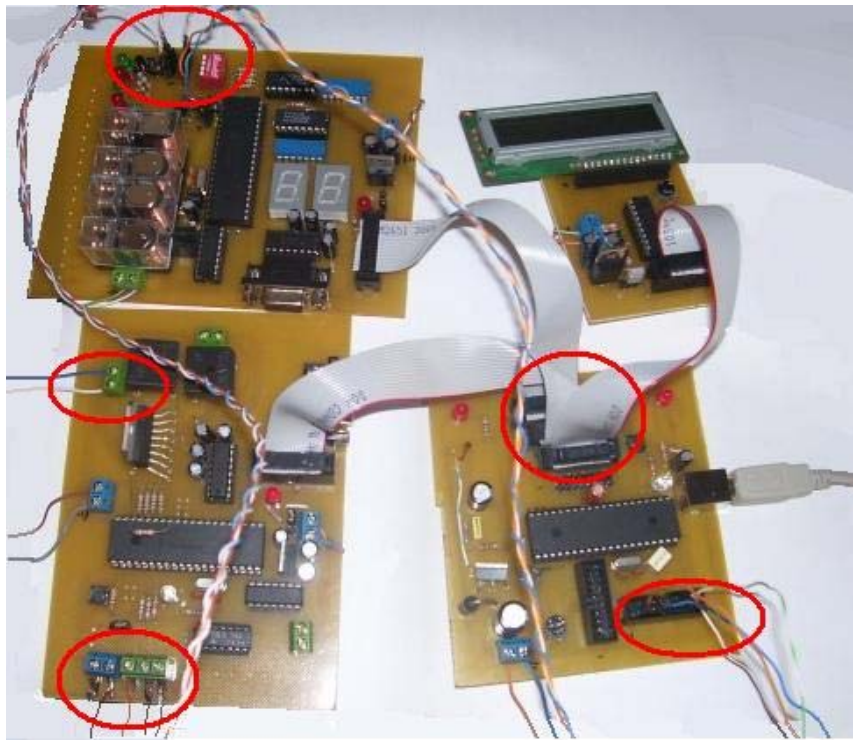


Figure 8.4 Connections

A sample GUI for this experiment made (Figure 8.5). In the GUI, there is a graphical control for displaying time based status of optical sensor. There are selection controls for setting the track regions to be read. And there is graphical control for displaying the sector based data information. There are also controls for manipulation the DC motor, RC servo motor, seven segment display and for displaying the status of optical sensor, switch, encoder count.

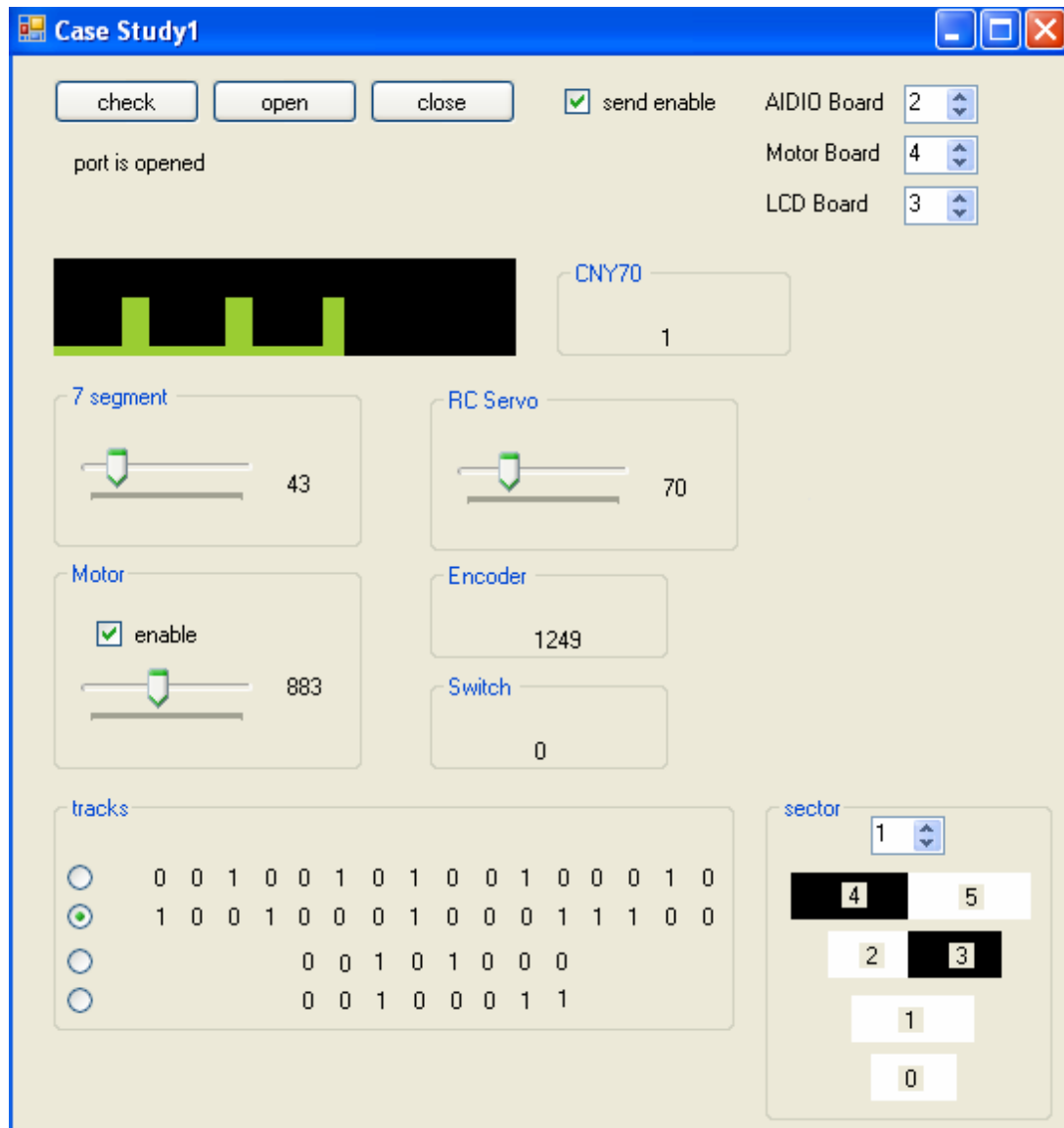


Figure 8.5 Experiment GUI

First the disk is rotated nearly at a constant speed at 40% PWM value. Then RC servo motor positions the optical sensor to a proper position. When switch is activated, the data collection from optical sensor is started. Fragmentation of the data is made by the help of encoder reading.

Then the speed of the disk is increased up to some wrong reading to be obtained. Nearly at 68% PWM value wrong readings obtained. The results are shown at Table 8.1.

Table 8.1 Experiment Results

real data	read data	real data	read data	real data	read data	real data	read data
track set 1		track set 2		track set 3		track set 4	
0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0
1	1	1	1	0	0	1	1
0	0	0	0	1	1	0	0
0	0	1	1	0	0	0	0
0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0
1	1	0	0	1	1	1	1
				1	0	0	0
				0	0	0	0
				0	0	1	1
				1	1	0	0
				0	1	0	0
				1	1	0	0
				0	0	1	1
				0	0	0	0

From the result it is seen that some erroneous readings occur at the third track set. This region is consisting of the smallest tracks on the disk. As a result sensor is reading this track region faster than other track regions.

CHAPTER 9

CONCLUSION

One of the start points of this thesis is the demand from the METU Mechanical Engineering Department mechatronics related courses. The demand is a standard board(s) with accompanying easy to use software for mechatronic education which is covering the general concepts of the content and which allows future developments, modifications. This should be a tool for students, the resources of which are manageable at all levels.

A literature survey is made about mechatronic courses and relevant products in the market. First design constraint adopted is that, the final system should cover the present course contents of courses offered at the ME department of METU. In addition to this, course survey is made to compare the resources needed at a mechatronic course and also for justification of using such a standard system on courses. After that products at the market which are using for mechatronic education are investigated. The market survey shows that some of these products might be a good choice at some conditions but because of their costs, maintainability and accessibility using these products does not seem proper.

As a result of the literature survey, the system architecture is determined. The survey reveals that various resources have to be connected to the computer. To manage this complexity, a modular approach is adopted where a single master, multiple slave configuration is used. For PC interfacing USB, for inter-boards interfacing SPI protocols are used. Boards are designed to complement each others with their specific features yet they can as well be used as stand alone.

At the heart of the system USB Board is placed. This board is responsible for PC communication and all the peripheral boards are connected to this board at default usage mode. Three peripheral boards are designed, namely, AIDIO, Motor and LCD boards. AIDIO Board is designed for general I/O manipulation applications. Motor Board is responsible for driving and controlling DC and stepper motors. LCD Board is designed for display purposes especially to facilitate standalone development and deployment. At the end of the thesis work, all of these boards are designed, manufactured and a firmware is written for each of them.

According the hardware resources available on the developed boards communication protocols are designed. According to these protocols, firmware of the boards are implemented and a DLL is written on the PC side which acts as the device driver for these boards. A sample GUI is developed on the PC side in order to demonstrate/teach the usage of the DLL in custom program development.

For demonstration of the overall system an experiment is prepared. In this experiment all the boards are used to show their integration capability. This experiment's lab sheet is prepared and sample results are presented.

9.1 Further Developments

For the future work, boards could be manufactured professionally at larger volumes. Reducing its manufacturing cost makes board more affordable by students.

USB Board's on board resources could be increased. AIDIO Board could be split into more specific little I/O boards. Motor Board could be modified to drive more motors. LCD board could be improved to drive graphic LCDs. New type wireless peripheral boards could be designed.

After long term usages and tests; at PC side and/or peripheral side other interfaces like PCIe, I2C could be implemented. High speed USB MCUs could be used instead of full speed ones.

At the firmware of the boards, new functions; at the DLL side, new methods could be implemented. More flexible protocol could be designed like dynamically changing bandwidth. Custom device drivers could be written at PC side.

Several experiments could be designed. Boards could be modified to be used at other kind of course experiments like fluid dynamics, vibration, HVAC course experiments.

REFERENCES

- [ME220] METU Academic Catalog,
https://catalog.metu.edu.tr/course.php?course_code=5690220 , last visited in June 2007
- [ME534] METU Academic Catalog,
https://catalog.metu.edu.tr/course.php?course_code=5690534 , last visited in June 2007
- [ME307] MECH307 - Mechatronics and Measurement Systems,
<http://www.engr.colostate.edu/~dga/mech307/index.html> , last visited in June 2007
- [Alciatore] Alciatore, D. and Hstand, M., "Mechatronics and Measurement Systems Course at Colorado State University," Proceedings of the 1994 Workshop on Mechatronics Education, Stanford University, July, 1994.
- [Ume] I. Charles Ume, Akio Kita, Sheng Liu and Sterling Skinner, "Graduate Mechatronics course in the school of mechanical engineering at Georgia Tech", Mechatronics Volume 12, Issue 2, March 2002, Pages 323-335
- [GIT Lab] Mechatronics at GaTech,
http://www.me.gatech.edu/mechatronicss_lab/ , last visited in June 2007
- [ENGG3400] Home Page for ENGG3400 Syllabus,
<http://www.soe.uoguelph.ca/webfiles/jzelek/engg3400/engg3400-syllabi.html> , last visited in June 2007
- [HandyBoard] The Handy Board, <http://handyboard.com> , last visited in June 2007

[IRIS] IRIS :: Introduction to Robotics & Mechatronics (151-0641-00) :: Syllabus, http://www.iris.ethz.ch/msrl/education/iris_intro/ , last visited in June 2007

[Trumper] David L. Trumper, Stephen J. Ludwick, "Development of 2.737 Mechatronics at MIT", Proceedings of the 1999 IEEUASME International Conference on Advanced Intelligent Mechatronics September 19-23, 1999

[MIT2.737] 2.737 Course Information - Spring 2006,
<http://web.mit.edu/2.737/www/> , last visited in June 2007

[ME333] ME 333 Course Information,
<http://lims.mech.northwestern.edu/~lynch/courses/ME333/2006/> , last visited in June 2007

[ME210] ME118: Intro to Mechatronics,
<http://design.stanford.edu/Courses/me118/me118.html> , last visited in June 2007

[Quanser] Quanser - Mechatronics Control Challenges ,
http://www.quanser.com/english/html/challenges/fs_chall_mechatronics_flash.htm
, last visited in July 2007

[Phidgets] Phidgets Inc. - Unique and Easy to Use USB Interfaces,
<http://www.phidgets.com/index.php> , last visited in July 2007

[W_FireWire] FireWire - Wikipedia, the free encyclopedia,
<http://en.wikipedia.org/wiki/Firewire> , last visited in July 2007

[Mikroelektronika] mikroElektronika | Solutions for the embedded systems,
<http://www.mikroe.com/en/>, last visited in August 2007

[NI] National Instruments - Test and Measurement , <http://www.ni.com>, last visited in September 2007

APPENDIX A

CD CONTENT

A compact disk (CD) is prepared for detail information of thesis work. The folders on the root are:

- 1_USB Board
- 2_AIDIO Board
- 3_Motor Board
- 4_LCD Board
- 5_PC
- 6_datasheets

At the board related folders; there are circuit layouts, Gerber files, schematics of boards under hardware folders and firmware codes with commented explanations under software folders. At 5_PC folder; there are drivers, boot loader program, source code of DLL and sample application with commented explanations. At 6_datasheets folder there are datasheets of items that used on boards.

APPENDIX B

SIMULATION OF HARD DISK READING LAB SHEET

B.1 Objective

In this lab you will read data from a rotary disk setup representing a hard disk. Interfacing of sensors, processing these data and communication between the Boards will be covered. There are a reflective optical sensor, a switch and an encoder as sensors on the setup. Collected data is displayed on the Boards and at the PC. Also speed control of disk and position control of a remote controlled (RC) servo motor has to be implemented. In this lab USB, AIDIO, LCD, Motor Boards will be used.

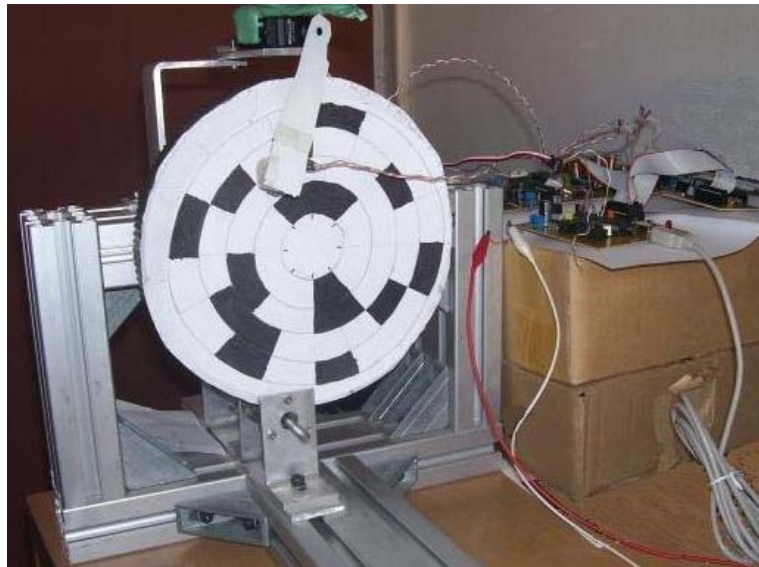


Figure B.1 Experiment setup

B.2 Procedure

B.2.1 Interfacing Reflective Optical Sensor and Switch

At this stage Reflective Optical Sensor and switch will be interfaced to the ADIO Board. CNY 70 will be used as reflective optical sensor. The detector (black dot) detects the reflected IR beam generated from emitter (blue dot) and sends a signal.

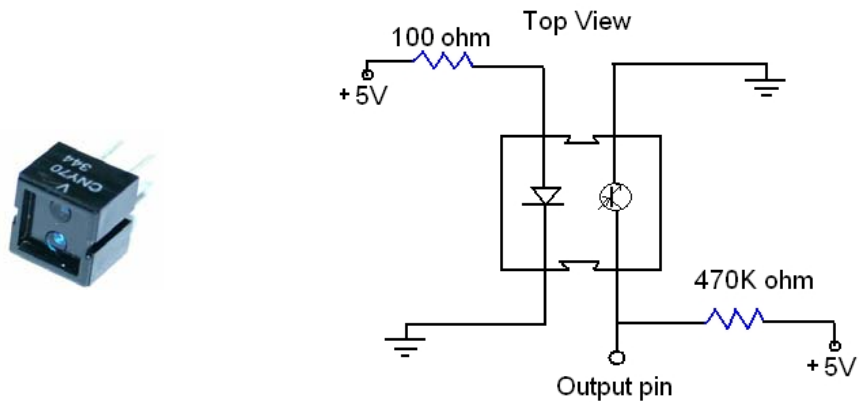


Figure B.2 Reflective optical sensor CNY 70 and its application circuit

Construct the circuit at right side and observe its behavior at different surfaces measuring output voltage from output pin.

Because of variable analog output of CNY70 a Schmitt trigger interfacing is needed. By this way we can get TTL signal for ADIO Board. As Schmitt trigger 74HC14 is used.

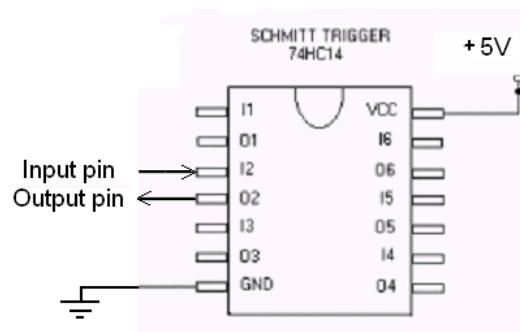


Figure B.3 Interfacing circuitry of CNY 70 to Schmitt trigger integrated circuit

Connect output pin of the Figure B.2 to the input pin of the Figure B.3. After integration of Schmitt trigger, observe the output voltage at output pin at the same surfaces again.

Now it is time to interface our reflective sensor to the ADIO Board. Power and signal pins are connected to the proper pin headers as shown below figure. Also connect the switch to the proper pin headers.

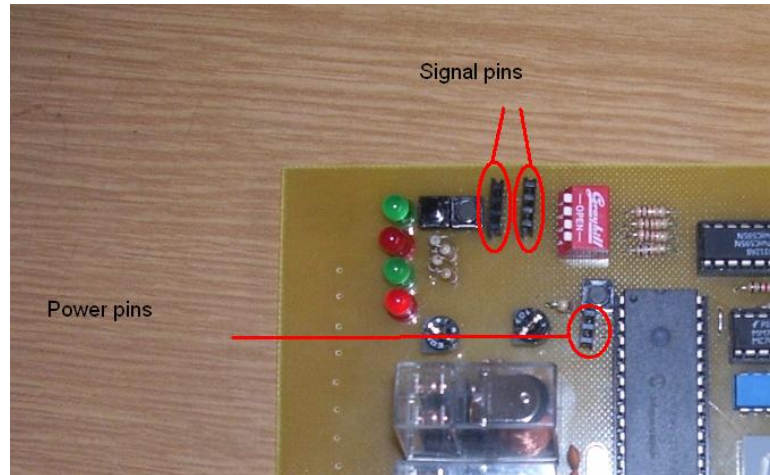


Figure B.4 AIDIO Board interfacing

After connections are done, PC interfacing will be done. Connect AIDIO Board to the USB Board. Connect USB Board to the PC.

Open .Net development environment at PC. Create a "Windows Application". Add references by locating "USBLibrary.dll". Initialize this DLL at under your form class

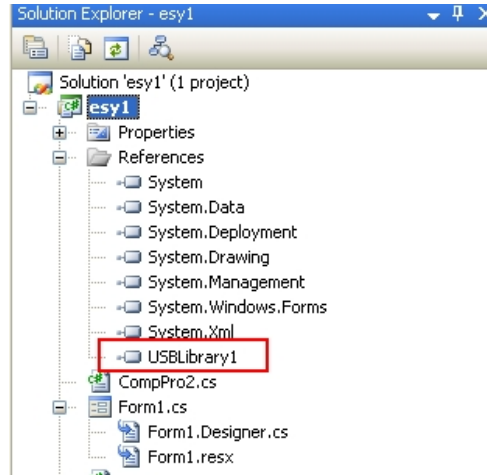


Figure B.5 Adding reference

```

namespace YourWindowApplication1
{
    public partial class Form1 : Form
    {
        private USBLibrary2.Class1 myLib = new USBLibrary2.Class1();

        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

Figure B.6 Initialization of DLL object

Now you can access DLL methods and create your GUI. Create a GUI that reads the value of reflective optical sensor and switch.

After successful sensor readings, interface these sensors to the rotary disk setup. You can get help from teaching assistant (TA) if you want at this level.

B.2.2 Counter

At this section rotation counter will be done by the help of a switch. The rotary disk is driven by hand and at every full turn switch send signal to the PC. This information will be displayed at GUI and at seven segments of AIDIO Board.

Before displaying numbers at the seven segments, a look up table has to be created. At this table there has to be byte values corresponding to displayed numeric character on seven segments. This table can be created by trial and error.

After creation of table, rotating rotary disk manually count the rotations and display the result at GUI and seven segments of AIDIO Board.

B.2.4 RC Servo Motor Control

RC servo motors are small DC motors with gears and built in controller circuits. The controller circuitry is tries to keep the shaft position at a desired position. The desired position is set by sending a PWM signal having a period in the range of at 10 to 25 ms.

The RC servo motor interfaced to USB Board's expansion connectors' proper pins as shown Figure B.7.

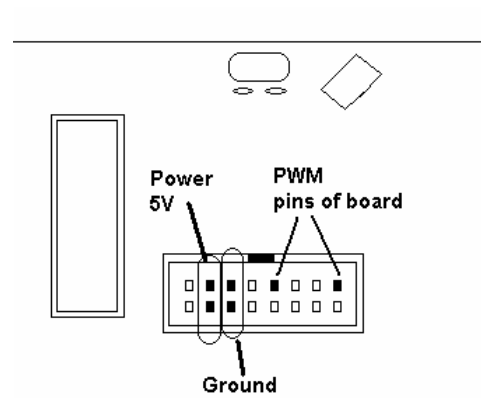
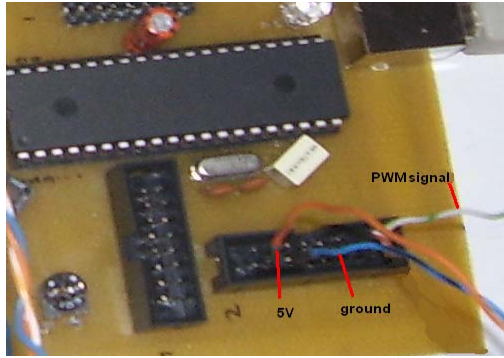


Figure B.7 USB Board interfacing

By sending a byte value to USB Board proper address you can play with RC servo motor shaft position. Before taking reading from the disk, RC servo motor has to place the optical sensor to proper positions. So first determine these positions and verify results.

B.2.3 Encoder Reading

At this section the encoder on the setup is interfaced to the Motor Board. Encoder readings will be displayed at GUI and LCD Board.

Connect power and signal cables of encoder to the related connections of Motor Board. Turn the disk manually and observe readings and display them on GUI and LCD Board.

B.2.5 DC Motor Driving

At this section, motor on the setup is driven by the Motor Board. Before driving ensure that all connections are correct. You can consult your TA.

Gradually increase the PWM value of motor. Try to drive the setup at low speeds. If constant speed can't be acquired implement a simple controller by getting feedback from the encoder. After driving at constant speed disk, collect data from reflective optical sensor with time tag and draw a graph at the GUI.

The hard disks are composed of sectors and tracks (Figure B.8). At this section of experiment the disk is driven at constant speed and RC servo motor is place the optical sensor to the first track section. When switch is off optical sensor data acquisition will start. By the help of encoder data, the read track data is segmented.

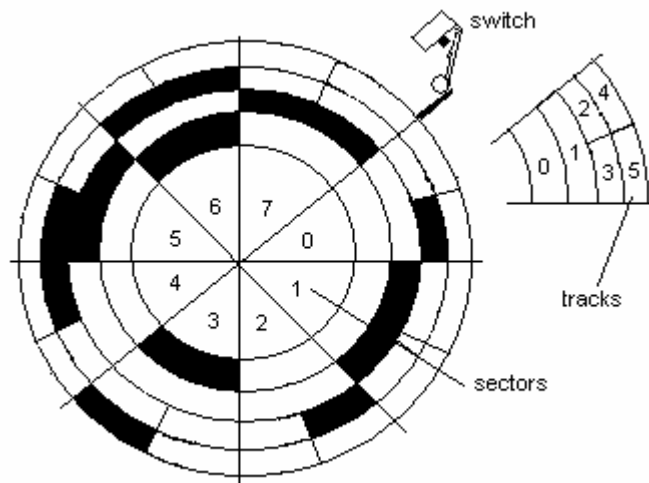


Figure B.8 Encoded paper

Then RC servo motor place the optical sensor to the next track section and read that track after switch is closed. At the end all the information on the disk is acquired as in track based format. Then convert this data into sector based one and compare the results.

Repeat this experiment at three different speeds of disk and compare accuracy of results.

Table B.1 Table for results

		exp1	exp2	exp3		exp1	exp2	exp3		exp1	exp2	exp3		exp1	exp2	exp3
	speed				speed				speed				speed			
bit #	track 1				track 2				track 3				track 4			
1	0				0				1				0			
2	0				0				0				0			
3	1				1				0				1			
4	0				0				1				0			
5	0				1				0				0			
6	0				0				0				1			
7	1				0				0				0			
8	1				0				1				1			
9	-	-	-	-	-	-	-	-	1				0			
10	-	-	-	-	-	-	-	-	0				0			
11	-	-	-	-	-	-	-	-	0				1			
12	-	-	-	-	-	-	-	-	1				0			
13	-	-	-	-	-	-	-	-	0				0			
14	-	-	-	-	-	-	-	-	1				0			
15	-	-	-	-	-	-	-	-	0				1			
16	-	-	-	-	-	-	-	-	0				0			