



Introduction

As a general rule, mechanical engineers dislike and avoid anything that looks like electrical engineering. Increasingly, however, sensors, actuators and control systems sit at the heart of mechanical devices, whether it is to optimise behaviour or to monitor the structural integrity, and usually also to provide feedback to the operator.

This lab is a brief introduction to mechatronics, and the idea that it is possible to use electronics without being an electrical engineer.

Microchip's PICDEM Mechatronics board has a number of input sources: temperature and light sensors, two potentiometers (variable resistors) and four switches. It has 8 LEDs and two motors: a stepper motor, and a brushed DC motor (with three feedback sensors for speed control). The board also has a simple LCD display, and a serial port for communicating with external devices.

The microcontroller ('Programmable Interface Controller' – PIC) that controls the board is a PIC16F917. This is an 8-bit processor, so generally works with integer numbers in the range 0–255, which can be written with two hexadecimal numbers (usually prefixed with '0x'), e.g.:

Binary	Hexadecimal	Decimal
1111 1111	0xFF	$15 \times 16 + 15 = 255$
1001 1010	0x9A	$9 \times 16 + 10 = 154$

This PIC microcontroller has a number of peripherals, i.e., designed-in components which make certain tasks possible and other tasks much easier:

- analog to digital convertor (ADC): converts an input voltage to an integer number
- pulse-width modulation (PWM): converts an integer number to an output voltage
- LCD controller
- serial port controller

Only certain pins (marked '/AN0' etc.) can be used for analog input, and certain others (marked '/CCP1' etc.) for PWM output; most pins can be used for digital input or output, where the voltage is either high or low. On the mechatronics board, several pins are reserved (mostly for controlling the LCD display) – these pins have white boxes around the pin descriptions. VDD and VSS represent supply voltage and ground respectively.

Setup

See also the PDF file 'MechatronicsLabNotes.pdf' in the folder E:\Stage2-MechatronicsLab.

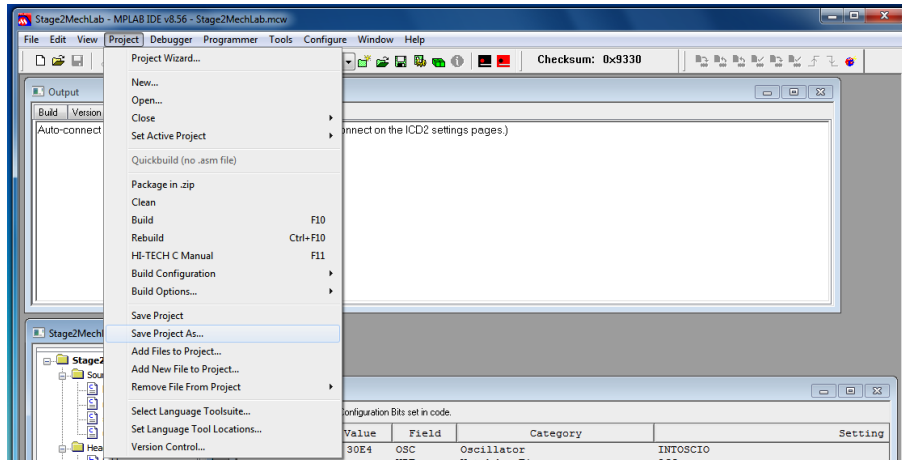
Connections:

- JP8 (near power supply): connect a **shunt** across the middle and right ('+5VDC') pins
- Grey wires from Brushed DC Motor: connect to screw terminals P9 ('DRIVE 1') and P10 ('DRIVE 2')
- Use **wire jumpers** to connect:
 - RA0/AN0 in Block J13 to POT1 in Block J4
 - RA1/AN1 in Block J13 to POT2 in Block J4
 - RD1 in Block J13 to SW4 in Block J4
 - RA6/T1OSO in Block J13 to J7 ('OPTICAL INTERRUPTER')
 - RD7 in Block J10 to LED D7 in Block J14
 - RD6 in Block J10 to LED D6 in Block J14
 - RD5 in Block J10 to LED D5 in Block J14
 - RD4 in Block J10 to LED D4 in Block J14
 - RC5/CCP1 in Block J10 to N2 in Block J1
 - RD2/CCP2 in Block J10 to P1 in Block J1
- Connect the serial cable from the computer to the serial port to the left of the LCD display.
- Connect the power supply to J9 (to the right of the LCD) – a green LED ('PWR ON') should switch on.

Note: If an orange LED ('FAULT') is on, press the 'CLR FAULT' button next to it.

Follow the instructions in the PDF 'MechatronicsLabNotes.pdf':

- Open the Stage2MechLab MPLAB Project file (on the **E:** drive) in the Stage2MechLab folder; in the MPLAB IDE **Project** menu use **Save Project As...** to save the project in a folder on your **H:** drive.



- Connect the ICD2 to your computer and then to the mechatronics board.
- Build the project.
- Program the PIC.
- Disconnect the ICD2 from the mechatronics board.
- Start the PuTTY terminal application to receive and log serial data from the mechatronics board.

Operation

The Stage2MechLab program is based on a clock that ticks every millisecond. The optical interrupt is checked every 1ms, but the analog inputs are checked only every 10ms (on different ticks of the clock). The outputs to the motor and LEDs are updated immediately after the corresponding analog inputs are read.

- The LCD should count from 0 to 59.9 in tenths of seconds.
- Turning Potentiometer POT1 should change the speed of the Brushed DC Motor.
- Turning Potentiometer POT2 should change the LEDs D4-7; these display 0-15 in binary.
- If Switch 4 ('SW4') is pressed, the values of analog inputs AN0, AN1, AN3 and AN4 are printed every 0.01 seconds.
- If Switch 4 ('SW4') is *not* pressed, the current time (0-999 milliseconds) and the value of AN0 are printed every time the optical interrupt is triggered.

Task

Turning Potentiometer POT1 changes the voltage at RA0/AN0 which is represented internally by an integer number in the range 0-255 (corresponding to input voltage range 0-VDD). This number is then used to generate a PWM-controlled voltage at RC5/CCP1 which is fed to the Brushed DC Motor. As the motor turns, the optical interrupter is triggered whenever the gaps in the motor wheel pass the IR light source; when the interrupt is triggered, the value of the analog input and the current time in milliseconds are written to the serial port.

- By importing the PuTTY session log into Excel, find the relationship between the input voltage (as a fraction of VDD) and the speed (in rev/s) of the Brushed DC Motor.
- Write a report (one side of A4, using a word processor and equation editor; and a separate Excel chart, properly and clearly labelled) explaining your experimental method and analysis. Give your result, and discuss anything unusual.

Remember to put your name and student number at the top, along with the date and module code.

Note

Time Difference: Since time is only given as the number of milliseconds, without the number of seconds, you need to check that the second time is larger than the first time, and otherwise to add one second to the second time. Make your life easier by learning how to use `=IF(, ,)` in Excel's formula bar.