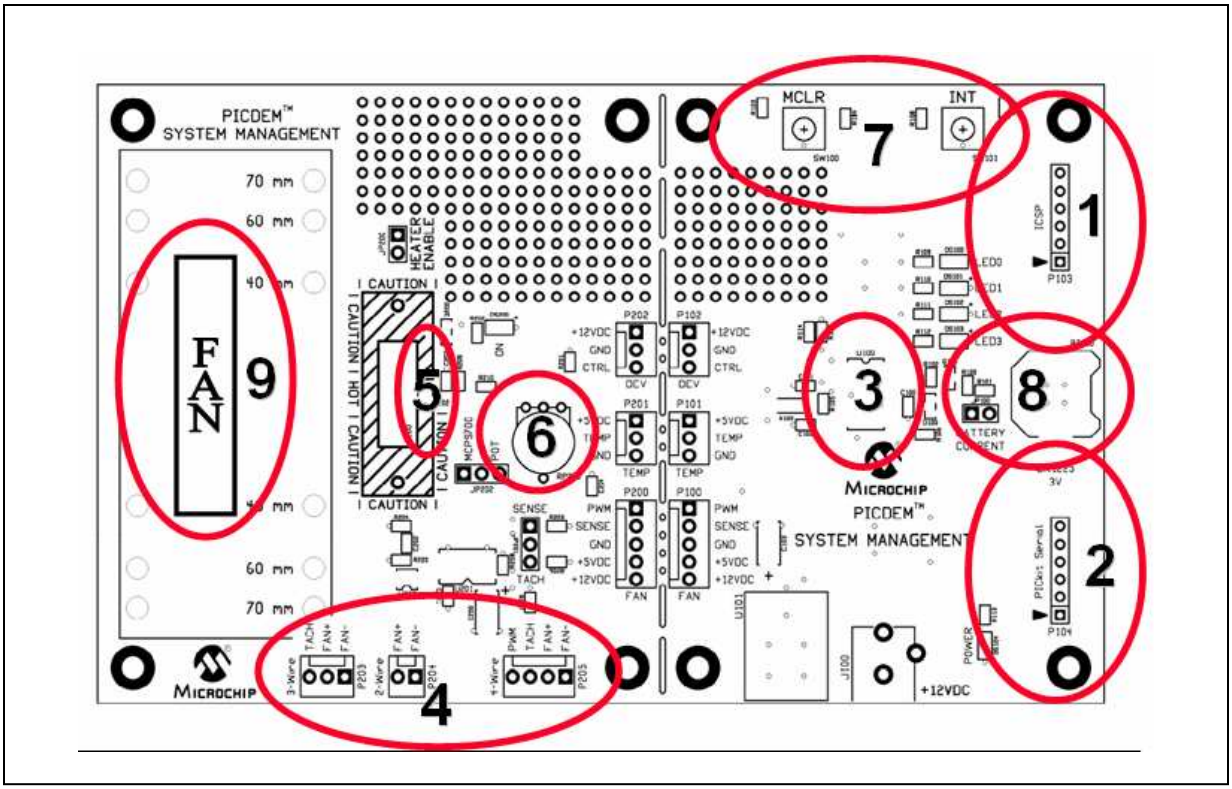
Introduction to the PICDEM System Management Lab

This lab is an introduction to the tools and software used in this course. This exercise will focus on the PICDEM System Management Board, PICkit Serial Analyzer, and System Management GUI.

PICDEM System Management

The PICDEM System Management Board is designed to emulate four independent, stand-alone devices on an I^2C bus. It emulates each device through the peripherals and firmware on a PIC16F886 Microcontroller. Step through the diagram below and inspect the board in front of you to understand each highlighted component.



1. PICkit™ 2 ICSP™ Connector – Programming header for PICkit™ 2 (with ICD-2 to PICkit™ 2 Adapter) programming or MPLAB ICD-2 programming

2. PICkit™ Serial Analyzer Connector – 6-pin header for PICkit Serial Analyzer. This header breaks out I^2C communications pins to the PICkit Serial Analyzer.

3. PIC16F886 Microcontroller – This microcontroller is preprogrammed with firmware that emulates four independent I^2C devices. It emulates a real

time clock calendar, serial EEPROM, analog-to-digital converter, and thermal fan controller. Throughout the course, we will learn about peripherals and software within the microcontroller that allows us to perform these tasks

4. Fan Connectors – The board provides terminals for four, three, and two wire fans. These terminals connect to the PIC microcontroller and allow the monitoring and control of each type of fan. In addition, the accompanying circuitry to measure the speed four, three, and two were fans is populated. Attached to the board currently is a three wire fan.

5. Temperature Sensor – The MCP9700 is an analog temperature sensor. This temperature sensor is connected to an analog-to-digital channel on the PIC microcontroller which allows the measuring of ambient temperature.

6. Potentiometer – The potentiometer and accompanying circuitry are designed so that as the potentiometer is turned, the voltage output range of the MCP9700 is swept. This allows all temperatures to be simulated.

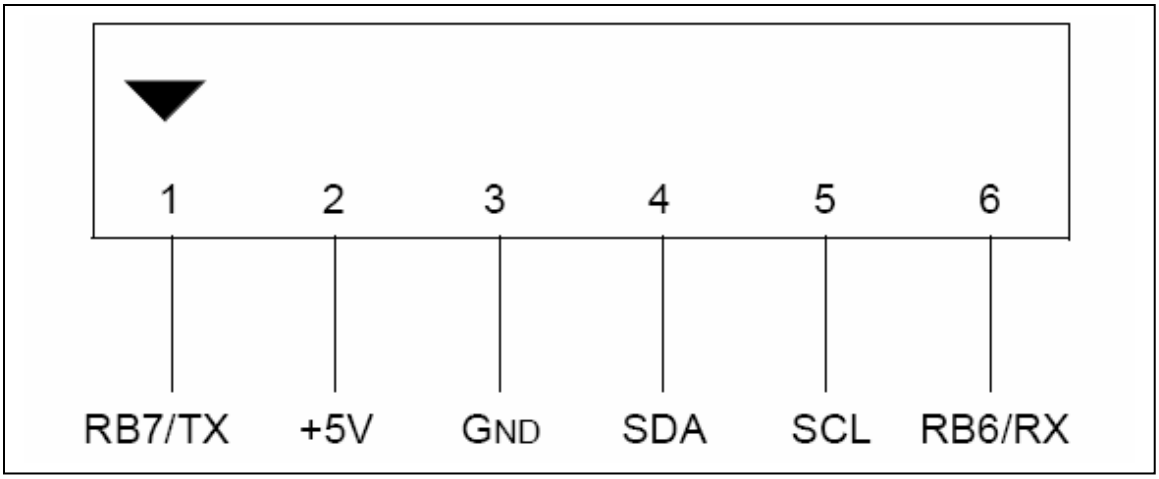
7. Push Buttons – The MCLR pushbutton resets the microcontroller, while the INT pushbutton allows for additional for use in further development.

8. Battery Backup Circuit – The battery backup circuit will be discussed the Real Time Clock Module. It ensures that power is always provided to the microcontroller. Power automatically switches from battery power to power supply power and vice versa when necessary.

9. Three Wire Fan – A three wire fan is connected to power, ground, and tachometer lines. The tachometer line provides speed feedback.

PICkit Serial Analyzer

The PICkit Serial Analyzer is a communications tool that can communicate to devices over a serial bus. The PICkit Serial Analyzer will be used in these exercises as a Master I^2C device that communicates to each of the emulated devices on the serial bus bus. Shown below is the pin-out of the device.



Lab Setup Procedures

[ ] Find JP200 labeled "Heater Enable." Make sure that JP200 is open.

[ ] Connect the +12V power supply to J100 labeled "+12VDC."

[ ] Verify that the fan is spinning and that the DS104 Power Led is on.

Each of the I^2C devices emulated on the microcontroller start in default states and is completely independent like similar stand alone I^2C devices. The fan is spinning because it has started in its default state and is trying to maintain a specific temperature.

[ ] Connect the PICkit Serial Analyzer to the PC using the provided USB cable.

[ ] Ensure that the "Power" LED on the PICkit Serial Analyzer is on.

[ ] Connect the PICkit Serial Analyzer to the PICDEM System Management Board connector P104 labeled "PICkit Serial."

[ ] Start the PICDEM System Management Software. (Click Start<Programs<Microchip System Management Board)

You will see five tabs. (Temperature Management, ADC, Real Time Clock, EEPROM, and I2C) Each of these tabs, with the exception of the I2C tab,

will communicate through the GUI interface to the PICkit Serial Analyzer and periodically poll the specific device indicated by the tab title. Data is shown in graphical form. In the next section we will see how we can manually read and write data to an I^2C device.

Using the I^2C Tab to read from I^2C Devices

The I^2C Tab is used to read and write to I^2C devices. It allows basic read and write operations with the ability to specify device addresses, data, and number of bytes to read or write. In this exercise, the I^2C Tab will be used to read the seconds data from an I^2C real time clock emulated on the PIC microcontroller.

[ ] Click on the I^2C tab.

[ ] Click on the Slave Address[W] field of the Read section of the tab and fill in the address section with the address "A2." 0xA2 is the I^2C Slave address of the real time clock device emulated within the PIC microcontroller.

Note: Clicking on the "x" in the text fields will change the "x" to a "d." This toggles the radix of the data entered into the text field. When an "x" is shown the value entered must be hexadecimal. When a "d" is shown the data entered must be decimal. For these exercises all values will be hexadecimal.

[ ] Click on the Word Address field of the Read section. This is the starting word address that will be read from the slave device. Fill this field with the value "02." 0x02 is the word address that corresponds with the real time clock, seconds data.

[ ] Note that the Slave Address[R] is automatically filled. This is the same slave address that was first entered, with the read bit set. This is an I^2C message format discussed in lecture. It is the combination read-write that sets the word address pointer, then reads data out starting at the word address location that is written first.

[ ] In the Byte Count section, fill in '1.' We intend to read out one data byte at address 0x02 of device 0xA2. Click execute to send the I^2C message.

[ ] The output window to the left should show a "Sent I2C Read Command" message in blue, which is the output from the PICkit Serial Analyzer. Shown in red is the data returned from the slave device. The value return should be a number between 0-59 since the data is seconds data. Try clicking execute multiple times to see how the data changes each second.

Using the I^2C Tab to write to I^2C Devices

In this exercise, the I^2C tab will be used to write data to a serial EEPROM emulated with the PIC microcontroller.

[ ] In the same I^2C tab, find the write section. This area is used to write data to I^2C devices.

[ ] Fill in the Slave Address[W] with the value “A8.” This will address the Serial EEPROM emulated on our microcontroller.

[ ] Fill in the Word Address with the value “10.” We saw in the previous section that this filling this field points to the word address that we intend to read. This has the same effect when writing. However, in the write field, it points to the word address that we intend to write. Address 0x10 is the word address location of the emulated EEPROM that we will be writing.

[ ] Fill the data field with the value “01.” You can write more than 1 byte by filling in the data fields that are stacked together. Click execute to send the I^2C message across the bus and write the data “01“ to address “10” of the emulated Serial EEPROM device.

[ ] On the output window, you will see a “Sent Write Cmd” message that indicates that the message was sent over the bus.

[ ] Use the Read command to read Device Address: “A8” at word location “10” for 1 byte to verify that the data 0x01 was written to address 0x10 of device 0xA8.

This concludes Lab 1. We have examined the PICDEM System Management Board to become familiar with its functionality. We have briefly highlighted how the PICkit Serial Analyzer is used in our application. Finally, we introduced the System Management GUI and how it works. The next labs will be more detailed and focus on each of the specific devices emulated on the PIC microcontroller.