University of British Columbia
Electrical and Computer Engineering
EECE 281/EECE282

Module 3 – SPI and Data Logging using Python

Introduction

Embedded systems are often designed to perform simple or repetitive tasks while connected to larger computers. Examples of such systems are found in many of today's computers: the mouse, keyboard, memory sticks, hard drive controllers, etc. For this module you will build one of such devices: an embedded thermometer using the DE2-8052 soft processor. The digital thermometer will serially transmit the temperature to the lab computer (or any computer with a serial port). You will program the lab computer (or any computer with a serial port) using either Matlab or Python to receive the temperature and conveniently present it in real time using a strip chart plot.

There are many free python distributions available. One that has all the functionality to complete this laboratory module is WinPython version 2.7.x.x available at:

http://winpython.sourceforge.net/

Please note that version 3.3.x.x may not be compatible with the scripts provided in this module.

References

A51 user manual included with the latest version of CrossIDE.

Using the MCS-51 Microcontroller by Han-Way Huang.

Python reference manual. Available online.

Pre-laboratory

1) Find the datasheet of the MCP3004 10-bit 4-channel ADC. Draw in your notebook its pin out and describe the function of each pin.

2) Find the datasheet of the LM335 integrated circuit. Draw in your notebook its pin out and explain why and how it is used. What is the temperature range of operation of the LM335?

3) What is the difference between a regular plot and a 'strip chart' plot in Matlab/Python?

Laboratory

1) **Testing the Serial Port of the DE2-8052.** Available in course web page you'll find the program 'hello.asm'. This program prints "Hello, world!" in HyperTerminal running in the lab computers[1] throughout the serial port of the DE2-8052. Compile, load, and test this program using the DE2-8052 soft processor and verify that you can receive the message through the serial port of a computer using HyperTerminal.

   Make sure you configure HyperTerminal to 115200 baud, 8 bits, parity none, 2 stop bits, and Flow Control 'none'.

2) **SPI communication.** Attach an MCP3004 ADC to the DE2-8052 soft processor. You'll need to communicate the ADC and the DE2-8052 using the Serial Peripheral Interface (SPI) standard. In order to do so, you'll need to write (or find!) an assembly subroutine to perform SPI communication. Write the subroutine into your lab notebook. Modify the 'hello.asm' program from the previous point so that it continuously prints the voltage from channel 0 of the MCP3004 ADC into HyperTerminal.

3) **Using Python to communicate with the DE2-8052 soft processor.** The Python script shown below opens the serial port in the host computer, constantly reads and prints a received value, and finally closes the serial port when CTRL+C is pressed in Python's command console. Attach an LM335 temperature sensor to channel zero of the MCP3004. Modify the program you wrote for the previous point so it converts the acquired value to temperature and transmits it through the serial port every second. To convert the voltage read from the LM335 sensor to temperature, a library of 32-bit arithmetic functions ('math32.asm') is available in the course web page for you to use.

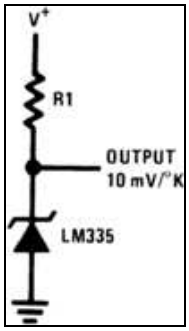
```
import time
import serial

# configure the serial port
ser = serial.Serial(
    port='COM1',
    baudrate=115200,
    parity=serial.PARITY_NONE,
    stopbits=serial.STOPBITS_TWO,
    bytesize=serial.EIGHTBITS
)
ser.isOpen()

while 1 :
    strin = ser.readline()
    print strin
```

   The script above assumes you are using COM1. For other serial ports, adjust accordingly. Also, Python expects a new line escape sequence ('\n') for each received value from the microcontroller. Connect the LM335 as shown in the figure below. Make V^+=5V and R1=2.2kΩ. To observe different temperature readings, you can **carefully** heat up the LM335 using the solder iron.

[1] HyperTerminal came installed with all versions of Microsoft's Windows up to Windows XP. To use HyperTerminal on Windows Vista or Windows 7/8, you can download an install a freely available version of HyperTerminal from the Internet.



4) Temperature strip-chart using Python. The script 'stripchart_sinewave.py' shows how to implement strip-charts in Python. A strip-chart can be used to plot the temperature transmitted from the DE2-8052 soft processor to Python in real time. Modify the provided script so it plots the data received from the serial port.

Demo the temperature strip-chart (in ºC) to you lab TA. Once again, you can use the solder iron to **carefully** heat the LM335 up! Copy into your notebook the assembly and Python programs; hint: the programs should be short! Also, save and print a copy of a good strip chart plot and attach it to your laboratory notebook.