



University of British Columbia  
Electrical and Computer Engineering  
ELEC291/292

## Lab 4: SPI, RS232, Temperature, Heart Rate, Strip Charts

Dr. Jesús Calviño-Fraga P.Eng.  
Department of Electrical and Computer Engineering, UBC  
Office: KAIS 3024  
E-mail: [jesusc@ece.ubc.ca](mailto:jesusc@ece.ubc.ca)  
Phone: (604)-827-5387

January 18, 2022

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Objectives

- Use SPI devices with the AT89LP51RC2 microcontroller.
- Use the serial port to connect the AT89LP51RC2 to a computer and interchange information.
- ELEC291: Measure temperature with the LM335. Display in strip chart.
- ELEC292: Measure heart rate. Display in strip chart.

**For this lab you can work with a partner.**

SPI, RS232, Temperature, Heart Rate

2

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## The Serial Peripheral Interface (SPI)

- Used to add multiple external I/O devices or functions to the microcontroller.
- SPI devices are **extremely** simple to connect to most modern microcontrollers/microcomputers.
- SPI devices are usually smaller and cheaper than their parallel bus equivalents.
- Many SPI devices available. Go to Digikey/Mouser and search for 'SPI'.
- SPI article in Wikipedia is excellent!

SPI, RS232, Temperature, Heart Rate

3

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## SPI Synchronous Data Communication Data Format

- Data length can be any size but optimized for 8 bits on most 8-bit microcontrollers.
- Clock is transmitted in a separate wire. Both the polarity of the clock and its phase are selectable.
- Data and clock are transmitted at the same time. Data is also transmitted and received at the same time!
- Most microcontrollers, including the 8051, can be setup as master or slaves to implement master/slave networks.

SPI, RS232, Temperature, Heart Rate

4

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Standard SPI Signals

- Master in / Slave out (MISO)
- Master out / Slave in (MOSI)
- Serial Clock (SCK)
- Slave select (SS')

SPI, RS232, Temperature, Heart Rate

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

5

## SPI Wiring modes

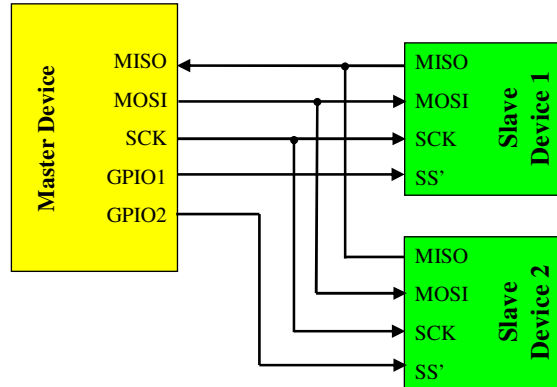
- Master: The device initiates and controls the flow of data. Can be configured as:
  - Multi-master.
  - Three wire single master.
  - Four (or more) wire master.
  - Four wire master with daisy-chained slaves.
- Slave: The device needs to be selected, and the clock is provided to it.

SPI, RS232, Temperature, Heart Rate

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

6

## SPI Wiring



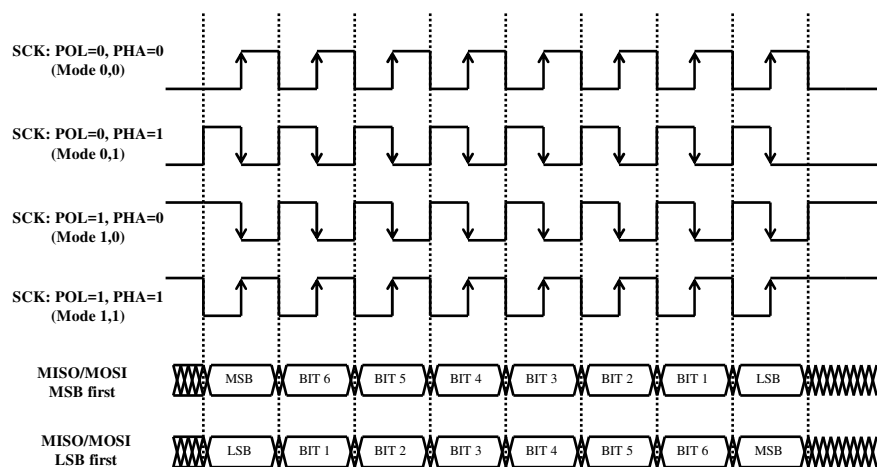
4-Wire Single Master and 4-Wire Multiple Slave Mode Connection Diagram. This is the most common configuration.

SPI, RS232, Temperature, Heart Rate

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

7

## SPI Timing

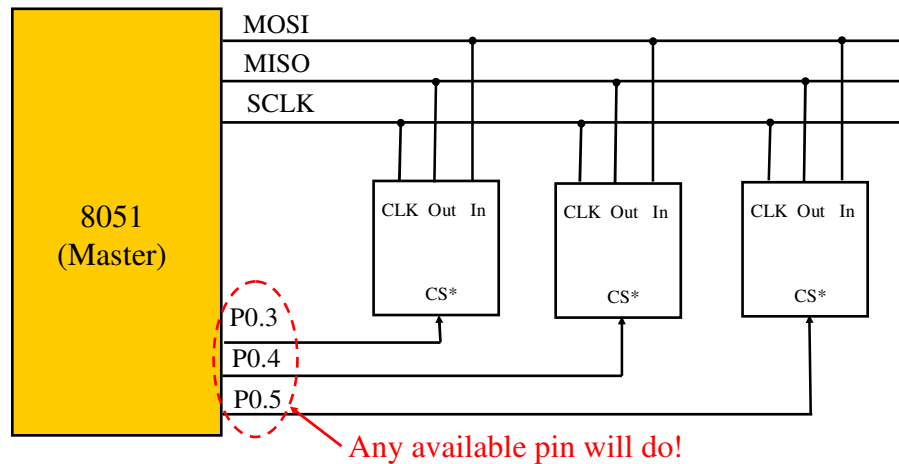


SPI, RS232, Temperature, Heart Rate

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

8

## Connecting Devices to the SPI port



SPI, RS232, Temperature, Heart Rate

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

9

## Bit Bang SPI

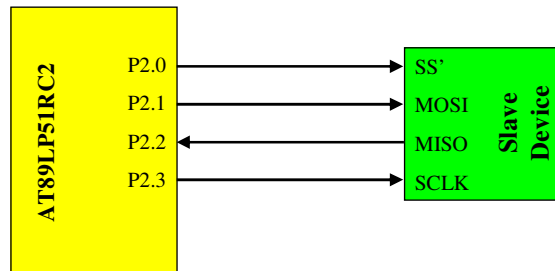
- Many modern microcontrollers have one or more built-in SPI controllers. This includes almost all Freescale (formerly Motorola), PICs from Microchip, 8051s derivatives from Silabs, Atmel, Dallas/Maxim, and NXP (formerly Philips), and ARMs from Atmel, NXP, and many others.
- Occasionally you may need to connect a SPI device to a microcontroller without an SPI controller. In this case we can **Bit Bang** SPI signals using GPIO pins!
- The AT89LP51RC2 has a built-in SPI controller but I used the SPI pins to In-System-Program (ISP) the  $\mu$ C, so... Bit Bang SPI it is!

SPI, RS232, Temperature, Heart Rate

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

10

# Bit Bang SPI



GPIO pins “arbitrarily” chosen... (they are actually the closest pins to the MCP3008 as you’ll see in the next slides)

SPI, RS232, Temperature, Heart Rate

11

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

# Bit Bang SPI in Mode (0,0)

```

#define ADC_CE P2_0
#define BB_MOSI P2_1
#define BB_MISO P2_2
#define BB_SCLK P2_3

unsigned char SPIWrite( unsigned char out_byte)
{
    // In the 8051 architecture both ACC and B are bit addressable!
    ACC=out_byte;

    BB_MOSI=ACC_7; BB_SCLK=1; B_7=BB_MISO; BB_SCLK=0;
    BB_MOSI=ACC_6; BB_SCLK=1; B_6=BB_MISO; BB_SCLK=0;
    BB_MOSI=ACC_5; BB_SCLK=1; B_5=BB_MISO; BB_SCLK=0;
    BB_MOSI=ACC_4; BB_SCLK=1; B_4=BB_MISO; BB_SCLK=0;
    BB_MOSI=ACC_3; BB_SCLK=1; B_3=BB_MISO; BB_SCLK=0;
    BB_MOSI=ACC_2; BB_SCLK=1; B_2=BB_MISO; BB_SCLK=0;
    BB_MOSI=ACC_1; BB_SCLK=1; B_1=BB_MISO; BB_SCLK=0;
    BB_MOSI=ACC_0; BB_SCLK=1; B_0=BB_MISO; BB_SCLK=0;

    return B;
}
  
```

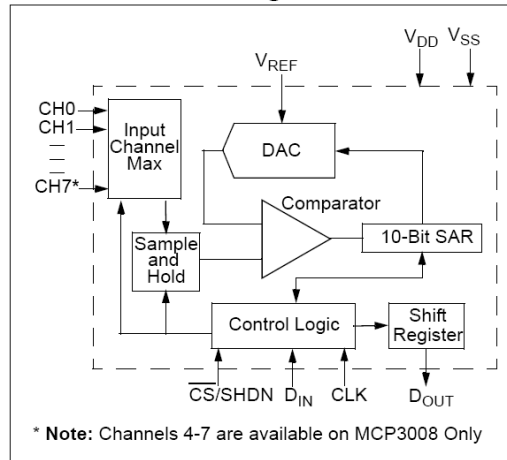
SPI, RS232, Temperature, Heart Rate

12

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Example: Using the MCP3008 10-bit, 8-channel ADC

Functional Block Diagram



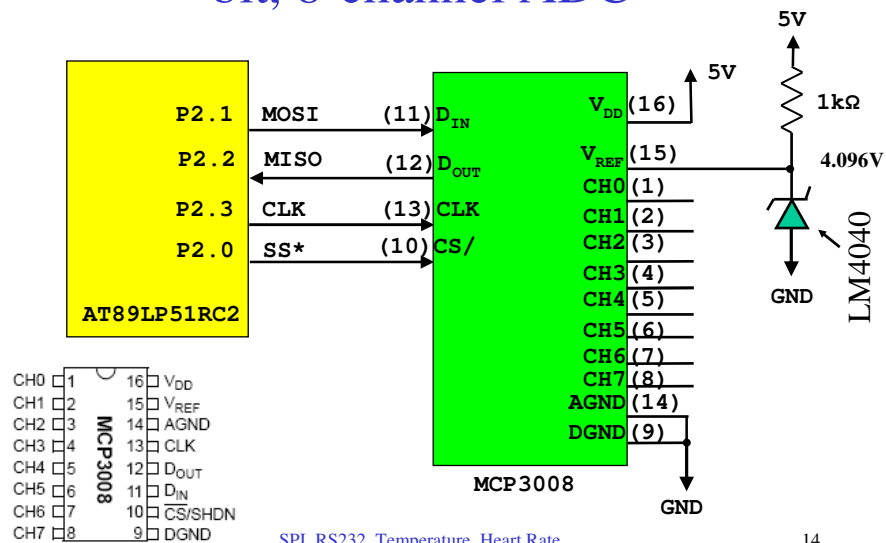
SPI, RS232, Temperature, Heart Rate

13

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

Figure taken From Microchip's MCP3008 datasheet

## Example: Using the MCP3008 10-bit, 8-channel ADC



14

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Voltage Reference LM4040DIZ-4.1

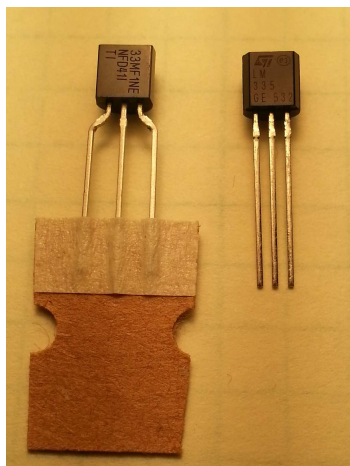
- Ideally we should use a good voltage reference for the “ $V_{REF}$ ” pin of the ADC. In your kit you have the LM4040DIZ-4.1: 1%, 150ppm/°C, 4.096V voltage reference. There are two ways of using the voltage reference:
  - We keep  $V_{CC}$  connected to  $V_{REF}$ , but connect the 4.096V voltage to one of the inputs of the ADC. With that input we can easily determine the true value of  $V_{CC}$ . The range of the other inputs is then nominally 0V to  $V_{CC}$ .
  - We connect the 4.096V directly to the reference input of the ADC. The range of the inputs is then 0V to 4.096V. This method seems to be the most sensible option for ELEC291/292.
- **WARNING:** Some old markings in the reference IC say “33MF1NE NFD41I TI” and has a piece of cardboard taped to its pins. Also, its package looks similar to that of BJTs.

SPI, RS232, Temperature, Heart Rate

15

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## LM4040 (reference) and LM335 (temperature sensor)



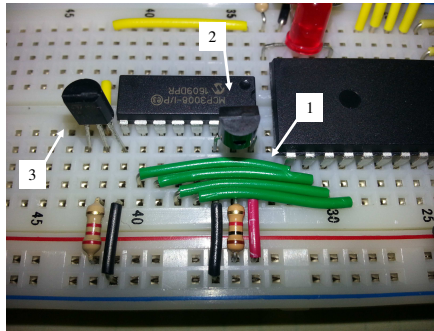
SPI, RS232, Temperature, Heart Rate

16

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.



## Wiring the Circuit (one possible way)



1. P2.0 (pin 20) to P2.3 (pin 23) connections to ADC.
2. LM4040DIZ-4.1
3. LM335 (un-adjusted) connected to channel 0.

SPI, RS232, Temperature, Heart Rate

17

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Example: Using the MCP3008 10-bit, 8-channel ADC

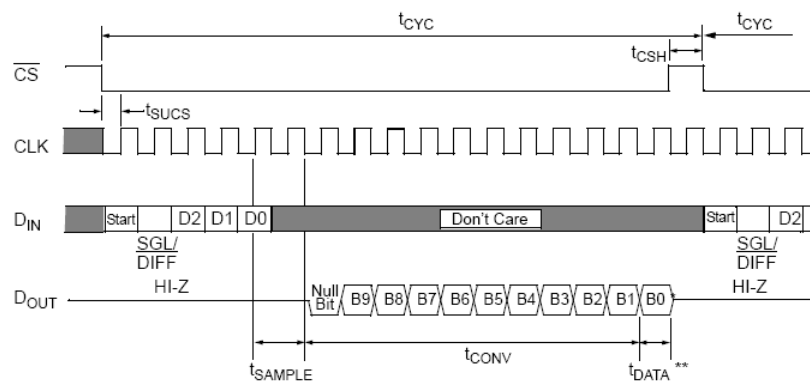


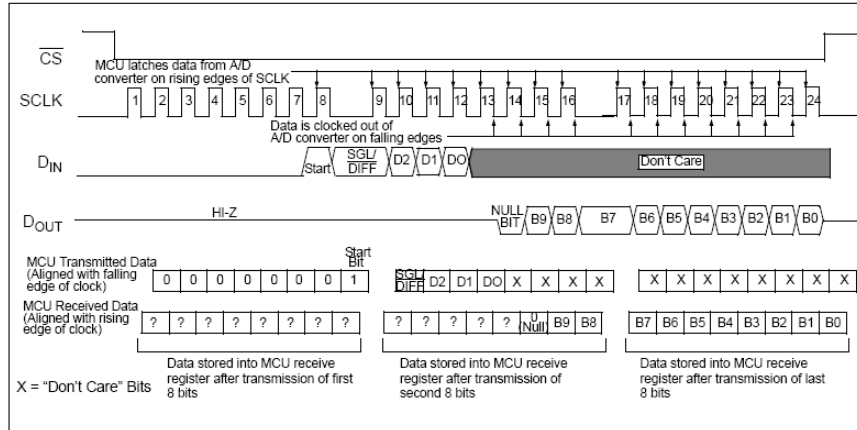
Figure taken from Microchip's MCP3008 datasheet

SPI, RS232, Temperature, Heart Rate

18

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Example: Using the MCP3008 10-bit, 8-channel ADC



**FIGURE 6-1:** SPI Communication with the MCP3004/3008 using 8-bit segments (Mode 0,0: SCLK idles low).

SPI, RS232, Temperature, Heart Rate

19

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

Figure taken from Microchip's MCP3008 datasheet

## Example: Using the MCP3008 10-bit, 8-channel ADC

```

/*Read 10 bits from the MCP3008 ADC converter*/
unsigned int volatile GetADC(unsigned char channel)
{
    unsigned int adc;
    unsigned char spid;

    ADC_CE=0; // Activate the MCP3008 ADC.

    SPIWrite(0x01); // Send the start bit.
    spid=SPIWrite((channel*0x10)|0x80); //Send single/diff* bit, D2, D1, and D0 bits.
    adc=((spid & 0x03)*0x100); //spid has the two most significant bits of the result.
    spid=SPIWrite(0x00); // It doesn't matter what we send now.
    adc+=spid; // spid contains the low part of the result.

    ADC_CE=1; // Deactivate the MCP3008 ADC.

    return adc;
}

```

SPI, RS232, Temperature, Heart Rate

20

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## The Serial Port

- The AT89LP51RC2 as well as most popular microcontrollers have one or more serial ports.
- The serial port uses the RS-232 communication standard. It was introduced in 1962!
- Perhaps the easiest way to communicate between a microcontroller and a computer!
- Unlike SPI, RS-232 is asynchronous: the clock is not shared between the processors.

SPI, RS232, Temperature, Heart Rate

21

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Asynchronous Data Communication Data Format

- A start bit used to synchronize the data. '0' or space.
- 5 to 8 data bits. For the standard 8051 the number of data bits is usually 8.
- Optional parity bit. Set or reset so that the number of ones transmitted is either odd or even. For the standard 8051 the parity is set to 'none' by default.
- One, one and half, or two stop bits. Always '1' or mark. For the standard 8051 is set to one stop bit.

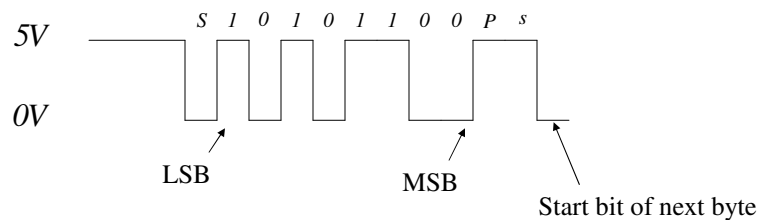
SPI, RS232, Temperature, Heart Rate

22

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Asynchronous Data Communication Data Format

- For example, transmit “00110101” using 8 bits, odd parity, one stop bit:



SPI, RS232, Temperature, Heart Rate

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

23

## Baud Rate

$$BR = \frac{1}{t_{bit}} \quad \text{Unit is 'baud'}$$

- Standard baud rates are: 110, 300, 600, 1200, 4800, 9600, 14400, 19200, 38400, and so on...
- The 8051/8052 with the correct crystal (For example 22.1184 MHz) can generate all the standard baud rates up to 115200 baud!

SPI, RS232, Temperature, Heart Rate

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

24

## “Magic Crystal” Frequencies

[https://en.wikipedia.org/wiki/Crystal\\_oscillator\\_frequencies](https://en.wikipedia.org/wiki/Crystal_oscillator_frequencies)

Frequency (MHz)	comm	UART	A/V	RTC	Primary uses
0.032				✓	Real-time clocks, watches; allows binary division to 1 kHz signal ( $2^5 \times 1$ kHz).
0.032768				✓	Real-time clocks, quartz watches and clocks; allows binary division to 1 Hz signal ( $2^{15} \times 1$ Hz); also low-speed low-power microcontrollers. Very common. Available as TCXO. <sup>[1]</sup>
					/180/150/110 kHz to both sides from the 10.7 MHz frequency)
11.0592		115200			UART clock ( $6 \times 1.8432$ MHz); allows integer division to common baud rates (96×115200 baud or $96 \times 96 \times 1,200$ baud); common clock for Intel 8051 microprocessors <sup>[18]</sup>
					Used in CD-DA systems and CD-ROM drives; allows binary
22.1184		460800			UART clock allows integer division to common baud rates up to 460,800(x16x3) or 921600(x8x3). Twice the 11.0592 MHz frequency.

SPI, RS232, Temperature, Heart Rate

25

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Serial port in the AT89LP51RC2

- “Hello, World!” example provided in Canvas.
- Download and install PuTTY  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- During PuTTY installation, use a folder WITHOUT white spaces in the name.

SPI, RS232, Temperature, Heart Rate

26

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Baud rate setup

- Using the baud rate generator (preferred in the AT89LP51RC2, as we can use the timers for something else):

$$BRL = 256 - \frac{f_{osc}}{16 \times baud}$$
$$BRL = 256 - \frac{22.1184MHz}{16 \times 115200} = 244$$

SPI, RS232, Temperature, Heart Rate

27

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## “Hello, World!” example

```
1 // AT89LP51RC2 "Hello, World!" example.
2 // ~C51~
3
4 #include <stdio.h>
5 #include <at89lp51rd2.h>
6
7 #define CLK 22118400L // SYSCLK frequency in Hz
8 #define BAUD 115200L // Baud rate of UART in bps
9 #if ((CLK/(16L*BAUD))>0x100L)
10 #error "Can not set baud rate because (CLK/(16*BAUD)) > 0x100 "
11 #endif
12 #define BRG_VAL (0x100-(CLK/(16L*BAUD)))
13
14 unsigned char _c51_external_startup(void)
15 {
16     AUXR=0B_0001_0001; // 1152 bytes of internal XDATA, P4.4 is I/O
17
18     // Configure serial port and baud rate
19     PCON|=0x80;
20     SCON = 0x52;
21     BDRCON=0;
22     BRL=BRG_VAL;
23     BDRCON=BRR|TBCK|RBCK|SPD;
24
25     return 0;
26 }
27
28 void main (void)
29 {
30     printf("Hello, World!\n");
31 }
32
```

SPI, RS232, Temperature, Heart Rate

28

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Serial port in C for the 8051

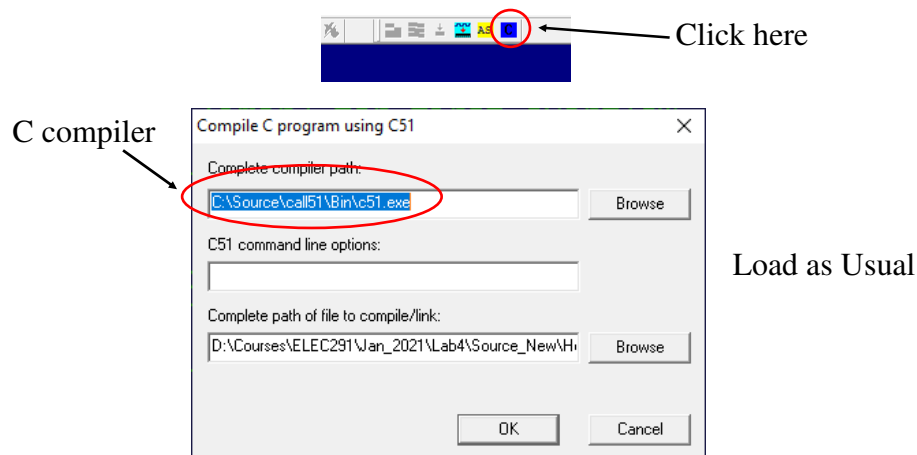
- putchar() and getchar() already defined in the standard libraries.
- printf() uses putchar().
- scanf() uses getchar(). (There is a bug in scanf() that prevents reading to a 'unsigned char' or 'char' variable properly. I don't know how to fix it even though I wrote the function! Lines 369 and 384 of scan\_format.c. The work around is to read to an integer variable.)
- Floating point arithmetic supported. Double not supported!
- Don't call scanf() or printf() from inside an interrupt service routine. EVER. Not even when you are using a fancy ARM or MIPS processor.

SPI, RS232, Temperature, Heart Rate

29

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Compiling with C



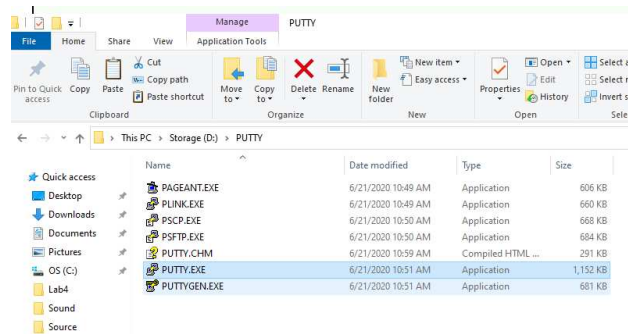
SPI, RS232, Temperature, Heart Rate

30

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

# PuTTY

- Download from here:  
<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
- Install in a folder **without spaces** in the name.

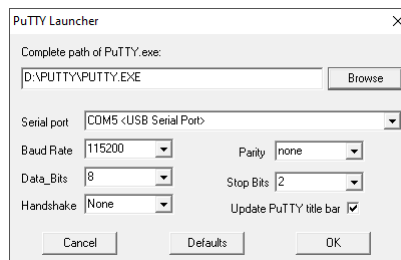
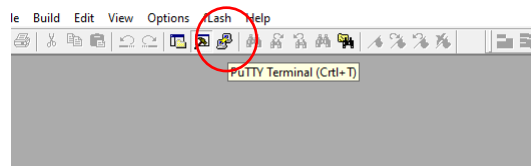


SPI, RS232, Temperature, Heart Rate

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

31

## Calling PuTTY from CrossIDE



SPI, RS232, Temperature, Heart Rate

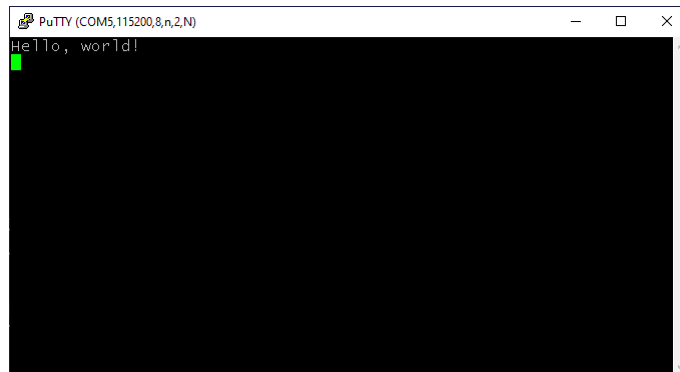
Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

32



# Running PuTTY

**Press the RESET** push button on the AT89LP51RC2:



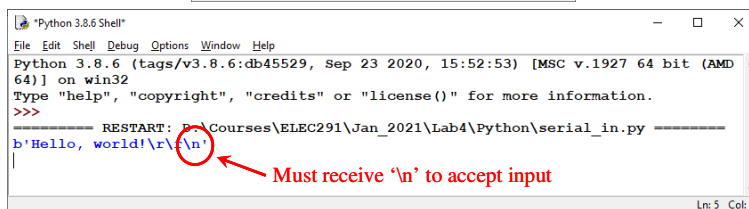
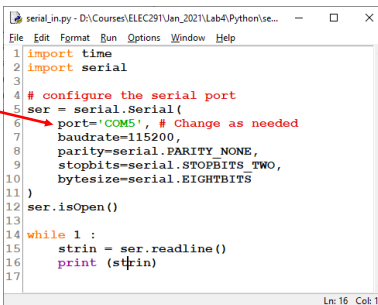
SPI, RS232, Temperature, Heart Rate

33

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

# Using the Serial Port with Python

Change to  
Match your  
computer



SPI, RS232, Temperature, Heart Rate

34

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Using the Serial Port with Python

```

1 import time
2 import serial
3
4 # configure the serial port
5 ser = serial.Serial(
6     port='COM5', # Change as needed
7     baudrate=115200,
8     parity=serial.PARITY_NONE,
9     stopbits=serial.STOPBITS_TWO,
10    bytesize=serial.EIGHTBITS
11)
12 ser.isOpen()
13
14 while 1:
15     strin = ser.readline()
16     print (strin.decode('utf-8'))
17

```

Different

```

Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: D:\Courses\ELEC291\Jan_2021\Lab4\Python\serial_in.py
Hello, world!

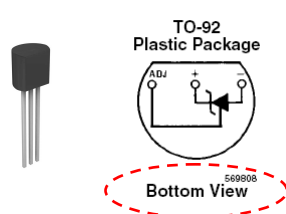
```

SPI, RS232, Temperature, Heart Rate

35

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## ELEC291: LM335 Temperature Sensor



For the LM335:

$+10\text{mV}/^\circ\text{K}$ ,  $-40^\circ\text{C} < t < 100^\circ\text{C}$

For the MCP3008 :

ADC: 10-bit,  $0.0\text{V} < V_{in} < 4.096\text{V}$

Un-calibrated temperature error: 2 to  $6^\circ\text{C}$

$$-40^\circ\text{C} = (273 - 40)^\circ\text{K} = 233^\circ\text{K} \rightarrow 2.33\text{V}$$

$$+100^\circ\text{C} = (273 + 100)^\circ\text{K} = 373^\circ\text{K} \rightarrow 3.73\text{V}$$

From the datasheet: "Included on the LM335 chip is an easy method of calibrating the device for higher accuracies. A pot connected across the LM335 with the arm tied to the adjustment terminal allows a 1-point calibration of the sensor that corrects for inaccuracy over the full temperature range." NO NEED TO DO THIS: leave the ADJ pin unconnected.

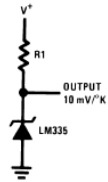
SPI, RS232, Temperature, Heart Rate

36

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## ELEC291: LM335 Temperature Sensor

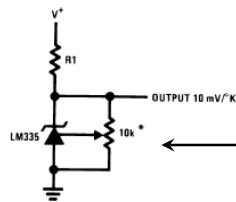
Figure 15. Basic Temperature Sensor



$R1 = 2k\Omega$  or  $2.2k\Omega$ , if you check the datasheet, most of the specs are @ 1 ma.

Figure 16. Calibrated Sensor

$V^+ = 5V$



NO NEED TO DO THIS.

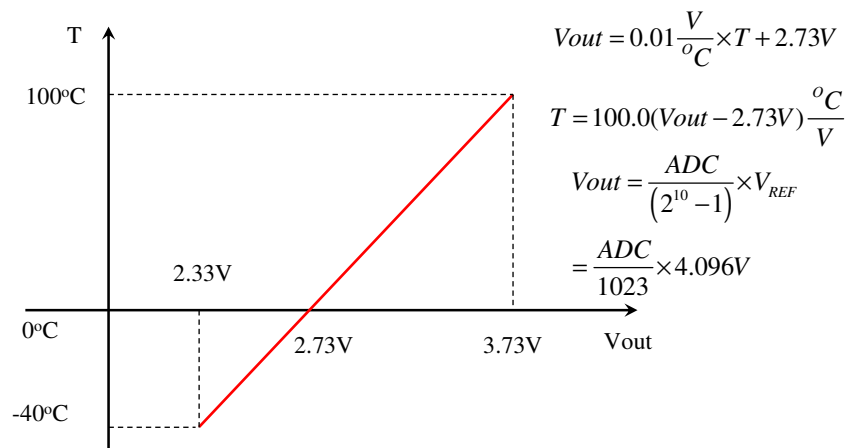
\*Calibrate for 2.982V at 25°C

SPI, RS232, Temperature, Heart Rate

37

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## ELEC291: LM335 Transfer Function



SPI, RS232, Temperature, Heart Rate

38

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Examples Provided

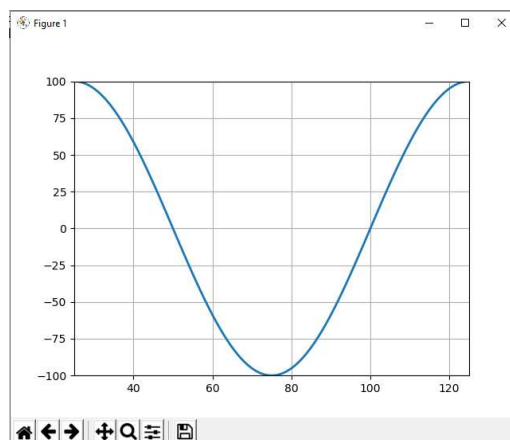
- For the AT89LP51RC2:
  - Hello\_AT89LP51RC2.c
  - ADC\_SPI.c
  - LCD\_4bit\_AT89LP51RC2.c
- Python:
  - stripchart\_sinewave.py
  - serial\_in.py

SPI, RS232, Temperature, Heart Rate

39

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Output of stripchar\_sinewave.py

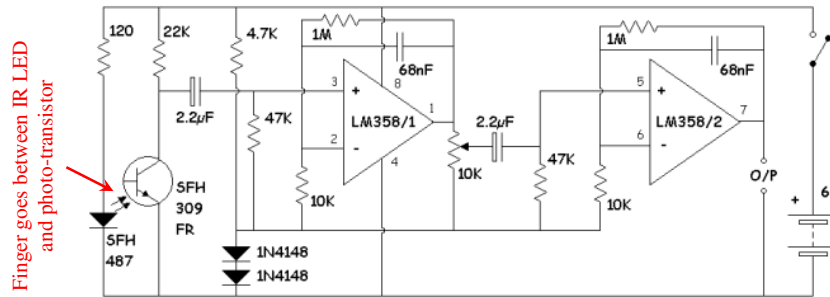


SPI, RS232, Temperature, Heart Rate

40

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## ELEC292: Heart Rate Sensor



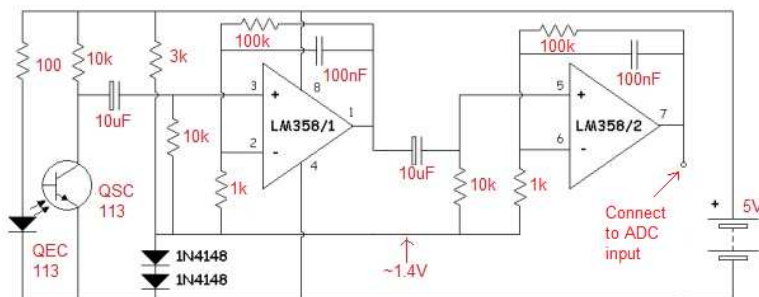
[http://www.picotech.com/experiments/calculating\\_heart\\_rate/](http://www.picotech.com/experiments/calculating_heart_rate/)

SPI, RS232, Temperature, Heart Rate

41

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## ELEC292: Modified Circuit



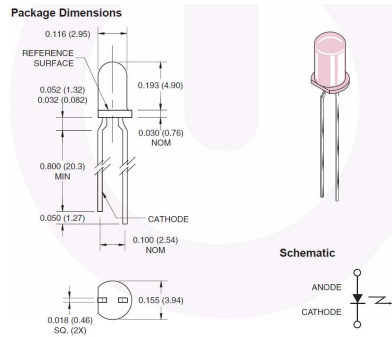
Using only components in your parts kit.  
Potentiometer removed. 5V power supply.

SPI, RS232, Temperature, Heart Rate

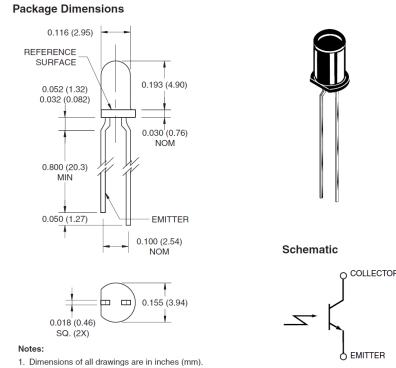
42

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

# IR LED and IR Photo-Transistor



QEC113



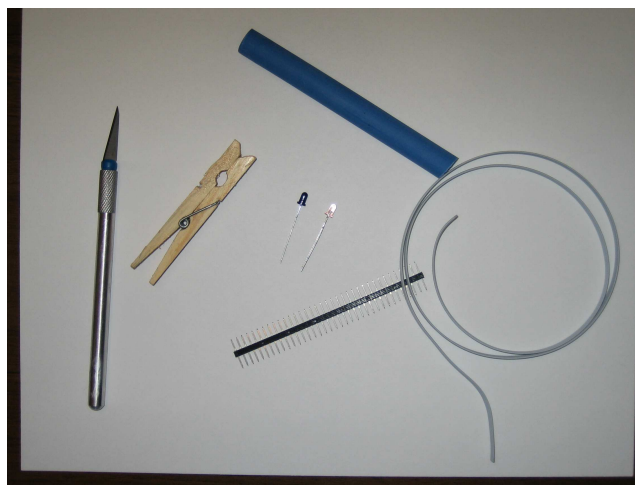
QSC113

SPI, RS232, Temperature, Heart Rate

43

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Finger Clip: before

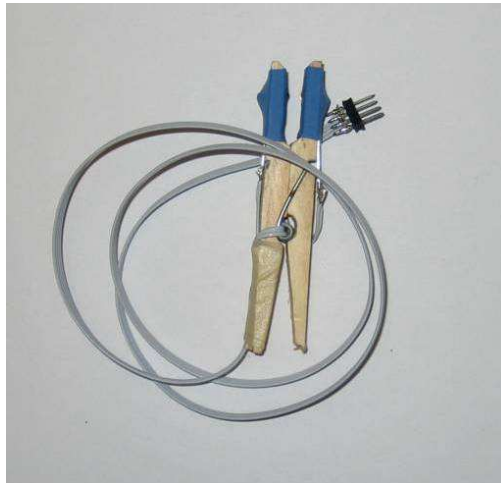


SPI, RS232, Temperature, Heart Rate

44

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Finger Clip: after



SPI, RS232, Temperature, Heart Rate

45

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.

## Final Remarks

- Support macOS:
  - WINE. Instructions on Canvas. Works both for ARM and Intel.
  - Virtual Machine (parallels or Oracle VB)
  - Bootcamp.
  - Compile remotely using EECE virtual machines and copy hex file to local machine.
- For Linux
  - Virtual Machine (Oracle VB)
  - WINE.

SPI, RS232, Temperature, Heart Rate

46

Copyright © 2009-2022, Jesus Calvino-Fraga. Not to be copied, used, or revised without explicit written permission from the copyright owner.