Student Numbers: PTLMUH012

HRLSUN002

Course: EEE3096S

Title: Prac 3 Report

Due Date: 18/09/2018

Q1

**a) Explain the SPI communication protocol with a timing diagram. (2)**

SPI is a serial synchronous type of communication. Two devices share a common clock signal to synchronise the exchange of data between them. SPI has the following message structure:

**Idle State -> Start Command -> Data Transmission -> Stop Command -> Idle State**

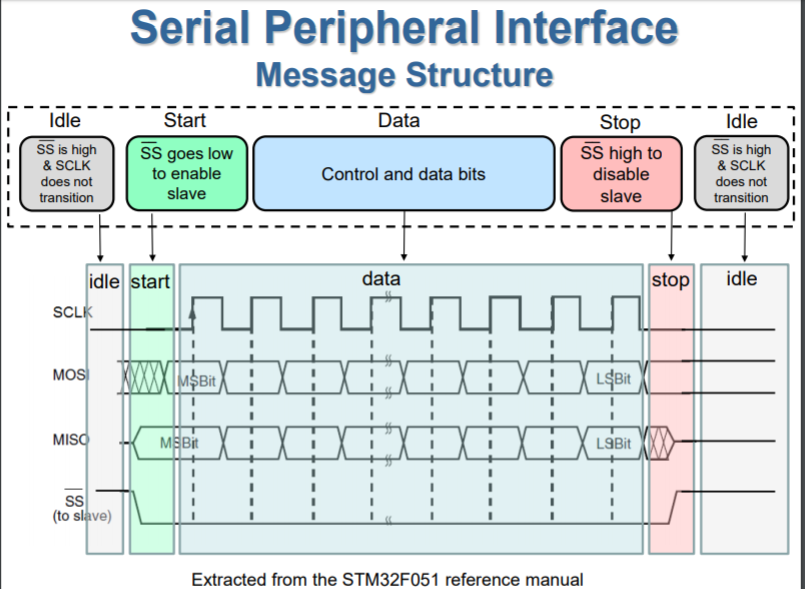
Idle State: SCLK does not transition and is high

Start Command: goes low to enable slave

Data Transmission: Control and data bits

Stop Command: high to disable slave

Idle State: SCLK does not transition and is high



**b) Define interrupt and threaded call-back in the context of an embedded system. (2)**

An interrupt is a signal that is sent to the processor by either hardware or software, indicating an event has occurred that needs immediate attention. Whenever an interrupt occurs, the controller completes the execution of the current instruction and starts the execution of an Interrupt Service Routine (ISR) or Interrupt Handler (IH). ISR/ IH tells the processor or controller what to do when the interrupt occurs. The interrupts can be either hardware interrupts or software interrupts

Multi-threading is the ability to run more than one thread at the same time, which means that you can go through more than one part of code simultaneously. So, for instance, your main program can be running in one thread, and when an event is detected in a second thread, it can communicate this back to the main thread. This is what call-back is about. Threaded call-back is therefore the ability to have a separate thread that “listens” for interrupts and has a set of instructions that will run when an interrupt occurs in the second thread.

**c) Write a function that converts a 10-bit ADC reading from the potentiometer to a 3V3 limited voltage output. (2)**

# Function to convert data to voltage level,

# rounded to specified number of decimal places.

def ConvertVolts(data,places):

volts = (data \* 3.3) / float(1023)

volts = round(volts,places)

return volts

**d) Write a function that converts a 10-bit ADC reading from the temperature sensor to a reading in degree Celsius (Have a look at the datasheet). (3)**

# Function to calculate temperature from MCP9700A data, rounded to specified

# number of decimal places.

def ConvertTemp(data,places):

temp = ((data \* 190)/float(1023))-40

temp = int(round(temp, places))

return temp

**e) Write a function that converts a 10-bit ADC reading from the LDR to a percentage representing the amount of light received by the LDR. The flashlight from a smartphone could be used as the maximum amount of light received by the LDR. (2)**

# Function to convert light data to percentage

# NB: the LDR decreases the voltage if you increase the amount of light shining on it, hence the reason for inverting the light\_level reading

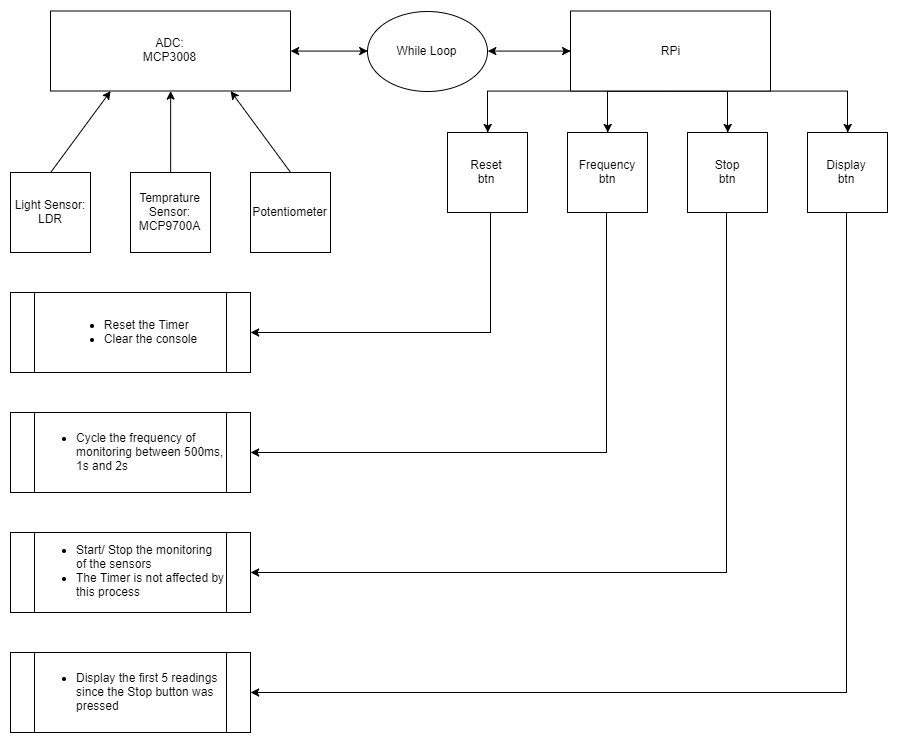
def ConvertPercent(light\_level):

inverted\_light\_level = 1023 - light\_level

light\_percent = (inverted\_light\_level/1023)\*100

light\_percent = int(round(light\_percent, 0))

return light\_percent

**f) Draw a flowchart of the system. (4)**