Practical 4

Introduction

The aim of this practical is to develop code for a simple environment monitor system. The system includes a temperature sensor (MCP9700A), a LDR (1K), a pot (1K), a MCP3008 IC and three switches.

Results

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

The SPI communication protocol can be either synchronous or asynchronous. The diagrams below show how the two communication protocols differ.

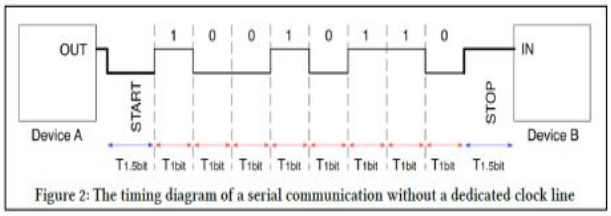


Figure 2: The timing diagram of a serial communication without a dedicated clock line

Asynchronous serial communication

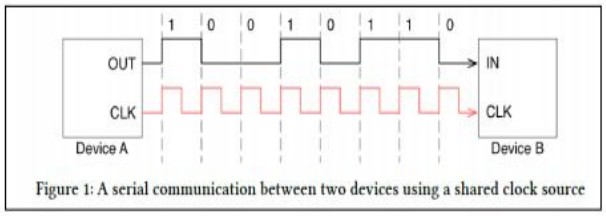
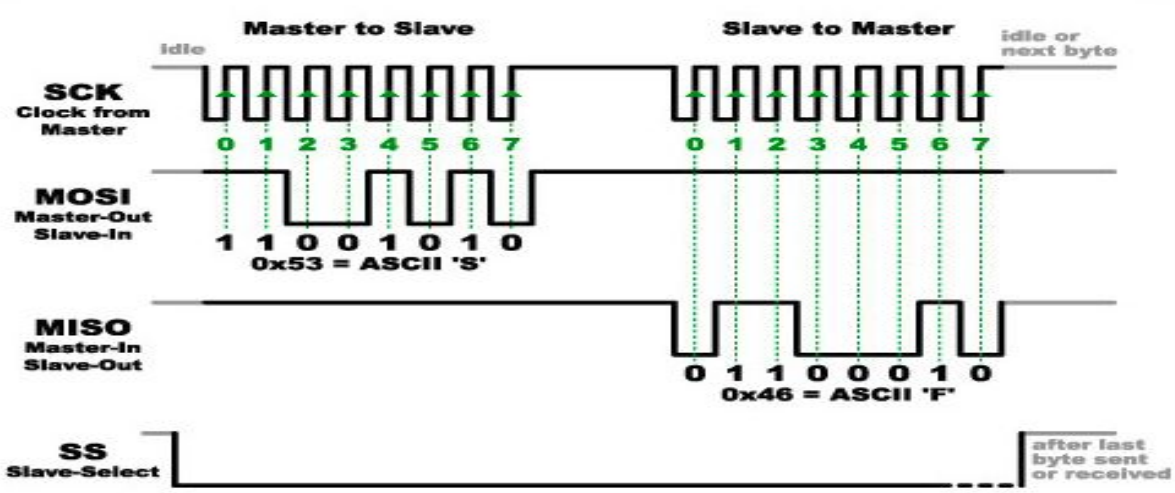


Figure 1: A serial communication between two devices using a shared clock source

Synchronous serial communication

The diagram below gives more insight on serial synchronous communication protocol which was the main focus in our lectures.



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

An interrupt is a signal that interrupts the processor, requiring it to stop immediately and handle the interrupt. A threaded call-back is a function or method that is invoked when a particular event (an interrupt) occurs.

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,
# places: number of decimal places needed
#first convert the data to a value between 0 V  and 3.3 V
#round the floating point number to places    number of decimal places
 #return the required value in volts
def ConvertVolts(data, places):
    volts =  (data * 3.3) / float(1023)
    volts  = round(volts, places)
    return volts.

```

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)

From the datasheet of the MCP9700A temperature sensor by Microchip technology, there is a linear relationship between the output voltage of the sensor and the temperature in degrees Celsius. This relationship is found from Figure 2-16 of the datasheet to be:

temperature_in _degrees_Celsius = 100*output_voltage – 50

```
# function to convert a 10-bit temperature value to degrees Celsius,
# places: number of decimal places needed
# first convert the 10 bit number to a voltage with two decimal places
# convert the output voltage to a temperature using the information from the datasheeet
# return the temperature
def ConvertDegCel(data, places):
    vout = ConvertVolts(data, 2)
    temp = ( 100*vout ) - 50
    return temp

```

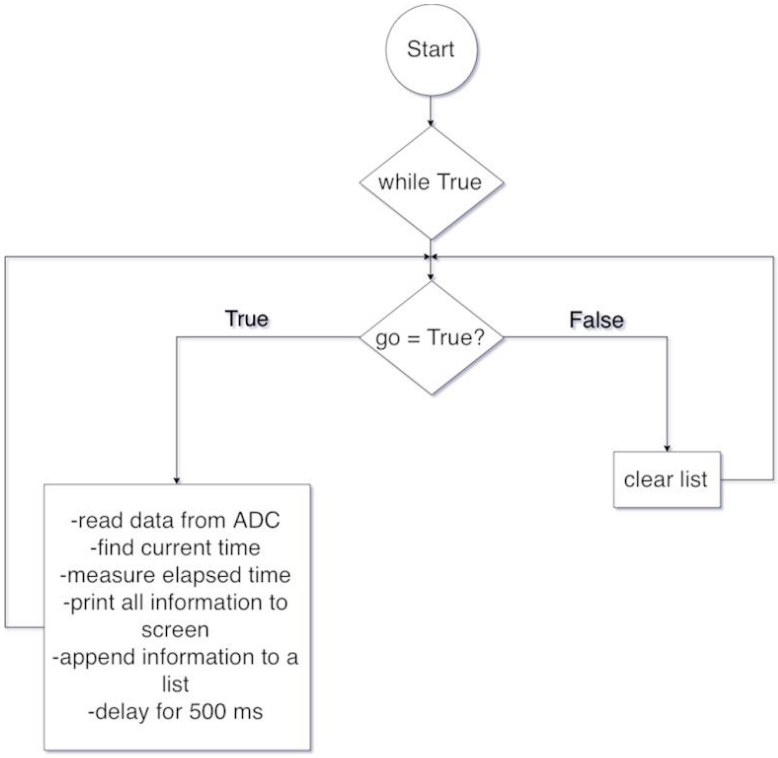
Write a function that converts a 10-bit ADC reading from the LDR to a percentage representing the amount of light received by the LDR. (2)The flashlight from a smartphone could be used as the maximum amount of light received by the LDR.

When the flashlight from a smartphone is used to shine light on the LDR, the 10-bit value read from the corresponding channel is 1023. This is taken as the maximum value that can be read from the LDR.

```
# function to convert light to percentage,
# places: number of decimal places needed
# first convert the value/light to a percentage
# round the value to places number of decimal places
# return the required value in percentage
def ConvertPercent(data, places):
    percent = (data * 100) / float(1023)
    percent = round(percent, places)
    return percent
```

-

Draw a flowchart of the system. (4)



CALL-BACKS (if interrupts received during execution)