Experiment 7: Communication Protocols in Embedded Systems

Due date : 27.12.2020 23:59

Kadir Ozlem
kadir.ozlem@itu.edu.tr

1 Introduction

In this experiment, our aim is to discover communication protocols such as UART, I^2C, SPI, CAN, etc. Please investigate these protocols and applications, compare these protocols in your reports. You can attain the design via this link.

- **You can not use block programming provided by Tinkercad.**
- **You can use all loop statement except do-while.**
- **You should add comments for necessary cases. Otherwise, you will get zero point.**
- **You can use digitalWrite and pinMode functions.**
- **You can not use delay function. Use millis or micros instead of delay.**
- **You can use analogRead() function to measure sensors value.**
- **You can not use map() function to transform sensor data to angle. You should write your normalization code.**
- **You can use "Wire.h" library for I^2C protocols.**
- **You can not use atoi, stoi, etc. functions to transform the Uart message to angle values. You should process it char by char.**
- **You can use "Servo.h" library to control servo engines.**

For your questions: Kadir Ozlem (kadir.ozlem@itu.edu.tr)

2 Part-1: UART Transmit Data

In this part, you will program the SensorMC (Sensor Microcontroller) whose design is presented in Figure 1. You will write a code that measures flex sensors and send the measurement results to the computer via the UART (Universal Asynchronous Receiver-Transmitter) protocol. Firstly, you should get the analog value of the sensors, then you will transform this information to the resistance value of the sensor and angle value of the sensor. You assume that flex sensors work linearly. Your sampling frequency must be 5 Hz.

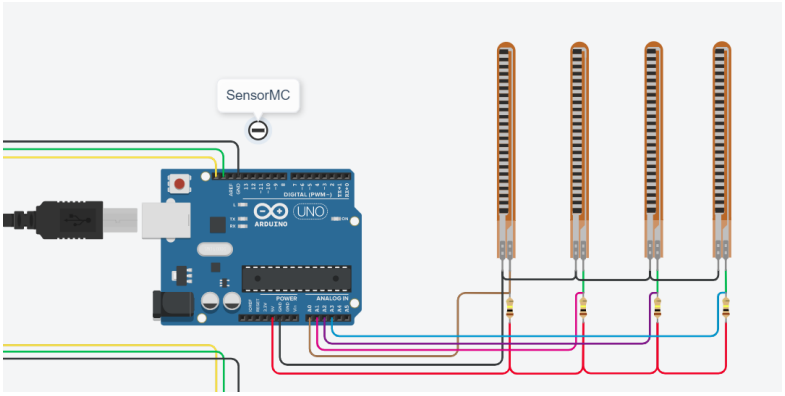

Figure 1: Circuit diagram of Part-1

Your output of each sensor must be "!SensorIndex;AnalogValue;Resistance;Angle#" for each sample. Table 1 shows example output for one sampling process.

!0;322;47.19;23#
!1;377;56.90;36#
!2;455;78.84;66#
!3;288;38.99;12#

Table 1: Example output for a sampling.

3 Part-2: UART Receive Data

In this part, you are going to implement the program that gets commands from the computer using UART protocols and change the position of the servo engines. You are going to use only ServoControllerMC in this part as shown in Figure 2. Your program can directly get the angle values of the servo engines from the computer and it will set the position of the servo engines using Servo.h library. The servo engines initial angle values must be zero.

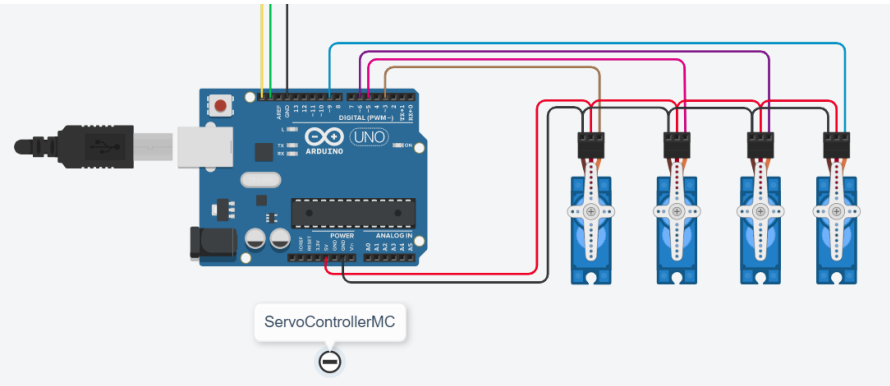

Figure 2: Circuit diagram of Part-2

Your program must send a success or failure message for each input value. Your input value starts with "!" char and ends with "#". Each servo engine value will seperated with ";". The input message has to contain angle information for each servo engine. The angle information should not be empty. The input message should not consist of a new line character. After an error occurs, you can ignore all characters until a new start or end character arrives. Table 2 shows the example input and output message. Your program must transmit the same output message for this input message.

Input Message	Output Message
!0;25;36;45#	S0:0; S1:25; S2:36; S3:45
!0;25;36#	Error! There is not enough input.
!0;25;36;45;23#	Error! There is extra input input.
!0;a5;45;23#	Error! Invalid Input.
!0;;25;36#	Error! All parameter must be filled.
!0;25;36;45!	Error! There is no end char.
!0;25;36;45#0;25;36;45#	Error! There is no start char.
!0;25;265;45#	Error! Angle value must be less than 180.

Table 2: Example output for received data.

4 Part-3

In the experiment, you are going to implement the program that gets angle values from flex sensors and control the servo engines using these angle values. Each flex sensor should control the servo engine under it. You can use the angle formula you implemented in the first part. The master microcontroller must control the data traffic. You should use the I^2C (Inter-Integrated Circuit) protocols for communication. It gets the angle values from the Sensor MC and sends them to the ServoControllerMC. You can use the same sampling frequency as in Part 1. You must program all microcontrollers as shown

in Figure 3

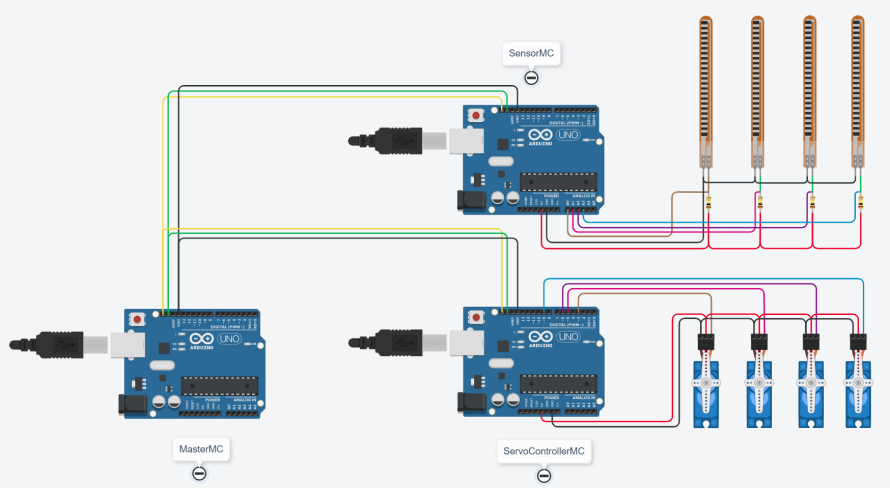

Figure 3: Circuit diagram of Part-3

5 Submission

- You should upload your experiment codes and report on Ninova, and please, do not send your experiment files via e-mail.
- You must upload each part's code seperately to the ninova.
- Your reports must be written in Latex format. The latex report template is available on Ninova. You can use any Latex editor whichever you want. If you upload your report without a Latex file, you directly get 0 as your report grade. You should upload both .pdf and .tex files of your report.
- You must collect the observed and calculated flex sensor data and compare them in the report.You should explain why two data is not equal.
- You should compare at least 4 communication protocols that are used in the embedded systems in the reports.
- Finally, please do not forget that late submissions are not accepted.