EE2800-01 Homework #5: Radio+Stepper

*Remember to follow the rules for homework submission, which were included with Homework #1.*

**Problem 1. Understanding serial communication**

The first problem is to help you understand serial communication. Suppose a serial stream contains the following sequence of bytes. Recall that a byte is represented by 8 bits or 2 hex numbers (4 bits each, called a “nyble” or “nibble”). A 4-bit number can represent a range of values from 0 to 24-1=15. An 8-bit byte represents a range of values from 0 to 28-1=255. Suppose we receive the following sequence of bytes from our radio:

0x03 0xE2 0xFF 0x17 **0xFF 0xFF 0x18 0x03 0xFE 0x00**

0xFF 0xFF 0x99 0x00 0xA3 0x03 0x10 0x21 0xFF 0x00

0x01 0xFF 0xFF 0x91 0x01 0xFF 0x00 0xFF 0xFF 0x8B

0x02 0x35 0x01 0x00 0x00 0xFF 0xF0 0x0F 0xDD 0xBB

**Part a: Determine all the x and y values that were transmitted. (They occur in pairs).**

To help you get started, we’ll do the first one. Recall that each packet has the following format:

The first two bytes indicate the beginning of a packet: 0xFF 0xFF

The next four bytes represent LSB of x, MSB of x, LSB of y, and MSB of y.

Scan left-to-right looking for a pair of 0xFF bytes, which indicates the beginning of a packet. The first packet is highlighted in bold for you. Combining MSB and LSB to form bytes gives the following: x = 0x0318 and y=0x00FE. If you type “convert 0x0318 to decimal” into google, you’ll find that x = 792, and 0x00FE corresponds to decimal y=254. The above sequence includes some “noise,” i.e., bytes that are not part of valid packets. These might originate from other radio transmitters causing interference. Just ignore them.

Note that the HC-12 program we discussed in class combines each LSB, MSB pair into one number using the formula:   
Val = LSB+MSB\*256. You can check this using the above results. For example, for x we have MSB=0x03 and LSB=18. Converting these to decimal gives MSBdec = 3 and LSBdec = 24. The total value is val=LSBdec+MSBdec\*256 = 24+256\*3 = 792 as expected.

**Part b:** Open the Arduino sketch “HW5Skel.ino”. This question asks about the “void checkHC12()” function. Assuming that the incoming byte stream is as shown above, the sequence of operations is: the first time called, the value of ‘state’ is 0, the value of ‘inVal’ is 0x03, the first if-else is executed and return #1 is used, and the final value of ‘state’ is 0, as shown in the table. Please fill in the remaining table values.

Initial state inVal return ln state

|  |  |  |  |
| --- | --- | --- | --- |
| **Initial value of state** | **inVal** | **Return #** | **Final value of state** |
| 0 | 0x03 | 1 | 0 |
| 0 | 0xFF |  |  |
|  | 0x17 |  |  |
|  | 0xFF |  |  |
|  | 0xFF |  |  |
|  | 0x18 |  |  |
|  | 0x03 |  |  |
|  | 0xFE |  |  |
|  | 0x00 |  |  |

**Problem 2. Joystick controlled wireless stepper motor and LED.**

Objective: This homework assignment will give you some practice using the HC-12 radio transceivers.

* This assignment requires that you work in pairs of 2 or 3. See the table below
* Build a transmitter system comprising:
  + **Software:** use “HC12packetXmt.ino.” No modifications are required.
  + **Hardware:**
    - HC-12 as discussed in class
    - Joy stick as discussed in class. Connect to A0 and A1.
* Build a receiver system comprising:
  + **Software**: modify HW5Skel.ino.
    - This already implements many, but not all, of the following requirements.
    - Required software modifications are indicated in UPPER CASE. Carefully read through HW5Skel.ino!
  + **Hardware:**
    - HC-12 as discussed in class
    - Stepper motor and controller board connected as discussed in class.
      * *Make this change: use Arduino pins 2, 3, 4, 5.*
      * Please put a piece of tape or paper on stepper motor shaft so it’s easy to see it turning.
    - The stepper motor will be controlled by **xval**, which has been received via the HC-12, and corresponds to the analog x value of the joystick. Your software will cause the following:
      * It will move clockwise if **xval**-**neutral** >**deadzone**
      * It will move counter clockwise if **neutral**-**xval** >**deadzone**

Where **neutral** is the analog joystick value when it’s not pushed,

and **deadzone** provides a region about neutral where the motor isn’t activated.

For example, if deadzone=50 and neutral=540, then joystick values in the range   
540-50 < **xval** < 540+50 will not cause any movement in the stepper motor.  
*Hint: use better choices for neutral or deadzone. These values are terrible!*

* + - One LED and resistor connected to pin 12. This will flash when the radio is communicating with the transmitter. Review and understand how this works in the skeleton code.
    - One LED and resistor connected to pin 9. We are receiving two values: xval and yval. We use xval to control the stepper, so what should we do with yval? We’ll make yval control the brightness of an LED. The code uses the ‘**analogWrite’** function and will be explained in class.

What to hand in:

* Each team hands in one report. Choose one team member to upload one set of files.
* A word document that lists of all team members.  
  It must include your group’s solution to Problem 1.   
  It should also have a brief summary of your results for Problem 2. If it worked as planned, just say that it works. If it doesn’t work, list the things that don’t work and your guess about what went wrong.
* Clear pictures of your transmit and receive breadboards.
* Your well documented Arduino sketch (i.e., hw5Skel.ino with your changes). Be sure to list team members in your Arduino Sketch comment section.
* A movie showing that your software and hardware accomplishes the above tasks. Specifically
  + When joystick is not being pushed, stepper doesn’t turn. LED9 at mid brightness
  + When joystick is forward or backward, stepper runs cw or ccw.
  + When joystick is left or right, LED9 is OFF or Full brightness.
  + LED on pin 12 should flash continuously indicating a good radio link
* **If you have a team member who is not participating, please report to Dr. Kubichek.**

**Homework #5 and Final Project Teams**

|  |  |  |  |
| --- | --- | --- | --- |
| Team | Radio  Channel | Member 1 | Member 2 |
| 1 | 005 | Shawn Dalke | Allison Davis |
| 2 | 010 | Owen Radcliffe | Nick Schaecher |
| 3 | 015 | Julian Duarte | Olanre Ebanks |
| 4 | 020 | Sam Bayles | Isaac Lloyd |
| 5 | 025 | Tony Nevshemal | Michael Kessesl |
| 6 | 030 | Will Mortimer | Elliot Jones |
| 7 | 035 | Kaden Zink | Zachary Churchwell |
| 8 | 040 | Josh Blaney | Dakota Julian |
| 9 | 045 | Zach Andriese | Arsene Bando |
| 10 | 050 | Isaac Hughes | Thomas Cole |
| 11 | 055 | Erick Arellano | Ahsan Javed, Daniel Hall |