ECE 206: STEMTera and Encoder

Challenge: STEMTera Digital Inputs and Encoders

In this lab, we'll follow up your work on Lab 2. Our goal is to design a circuit as well as the associated software to read and keep track of the position of the encoder using a STEMTera.

Prelab Deliverables: (due at the end of class on Compass)

- Consider your scope traces from Lab 2. How many times/second would you need to read the value of the A and B signals to keep track of the encoder position in the "slow" case? How many times/second in the "fast" case? Explain your answer based on what you recorded in your scope traces. (Note: You can assume each time index in Serial Plotter equals *x* seconds and leave your answers in terms of *x*.)

Challenge:

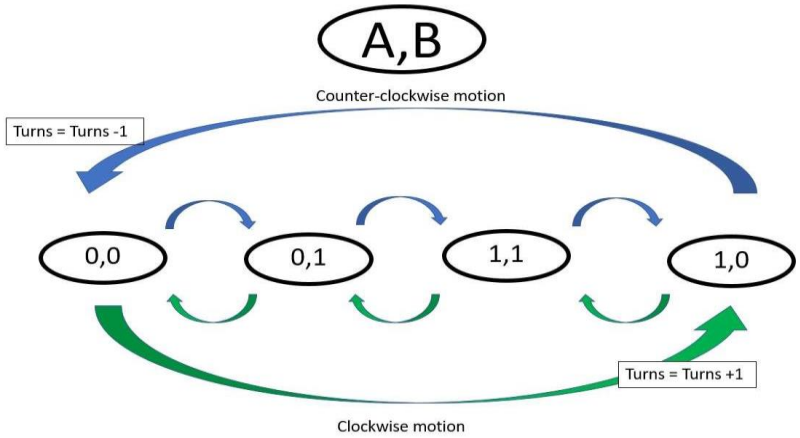
- Rebuild or otherwise construct the circuit you used in Lab 2 to read the A and B channels of the encoder into the STEMTera. Connect A and B channels to some STEMTera pins. Use an additional 7 pins to control the 7 segment LED (you won't need the decimal point) to display digits 0-9.
- Use the encoder to increment and decrement the digit based on knob rotation. Clockwise rotation should INCREASE the number shown (up to 9), and counter-clockwise rotation should DECREASE the number shown (down to 0). Demonstrate your working circuit to your TA.

Bonus Challenge: Use the second digit of the LED to make your circuit work between 0 and 99.

Hints: You should have created a flowchart after Lab 2 which can be used to decode A and B signals from the encoder to a count. The skeleton code for your design is provided as follows:

```
int total_rotation = 0; //this is a global variable all functions can access
int last_a = 0;

// one time run setup function
void setup() {
 // initialize digital pins 12/13 to be inputs (channels A and B respectively)
  pinMode(12, INPUT);
  pinMode(13, INPUT);
}

// infinite loop for real-time operation
void loop() {
  int a = digitalRead(12);//read pin 12, call it 'a'
  int b = digitalRead(13);//read pin 13, call it 'b'

 if ((last_a == LOW) && (a == HIGH)){
  //if A has gone from low to high
  //you should do something here depending on B
 }
 if ((last_a == HIGH) && (a == LOW)){
  //if A has gone from high to low
  //you should do something here depending on B
 }

 //display the new value here on the LED

   last_a = a; //update last_a
}
```

When plotted on a state diagram, the strategy of the code above looks something like the diagram below. (Note that clockwise/counterclockwise may be reversed in this diagram):



Report Deliverables:

- Include a diagram/sketch of you circuit.
- Include commented code for your design.
- Obtain your TA signoff on the circuit in operation
- Bonus challenge: relevant diagrams, sketches, code and explanations (Bonuses are optional)

Report should contain, at minimum the following elements:

- Title, Names, Dates
- Statement of Purpose
- Plan and Execution
- Results / Conclusion
    - Answer any questions posed in the lab challenge.
    - Explain your understanding of what happened and why.