Port Manipulation allows for faster low-level manipulation of the i/o pins on the Intel Galileo.

Download the PortManipulation Library provided to begin.

Import code for initialization:

```c
#include <PortManipulation.h>
define STARTING_PIN 2
PortManipulation port(STARTING_PIN);

void setup () {
  port.begin();
}
```

This library works for 8 bits/pins. The manipulation will start at STARTING_PIN and will end 8 after so in the case of ‘2’, it will end at pin ‘9’ including it.

The only function we will be using is portMan(unsigned char).

port.portMan(8); Will write in Binary B00001000 to the pins with the least significant bit starting at STARTING_PIN.

**Set-up**

Begin by wiring 8 LEDs from pins 2 – 9 of the Intel Galileo. Initialize port manipulation by using the code above and begin by testing different unsigned char values using the function port.portMan(#).
Video Demonstration: https://youtu.be/m0qhl33XxVM

PART I - Single synchSM
Exercise 1 - Bouncing LED: Capture a synchSM that lights the LEDs one-at-a-time in sequence: Pin 2, then Pin 3, then Pin 4, ..., then Pin 9, then Pin 2 again. Each LED stays on for 1 second. Next, have A0 select the sequence direction: 0 as above, 1 in reverse.

Exercise 2 - Pattern Sequencer: Next, create any sequence of 10 output patterns, such as 11111111, 01010101, 10101010, ... (ten such patterns). Store them in an array. Each pattern displays for 1 second. A0 being 0 goes through the pattern forwards, 1 backwards.

Exercise 3 - Music Accompaniment: Use a smart phone/tablet to play at least 20 seconds of a (clean) song of your choice, create a light show that accompanies the song, following the song in some way.

PART II
Using at least three "concurrent" synchSMs, devise a system where A1A2 being 00 executes the above Bouncing LED synchSM, 01 the Pattern Sequencer, and 10 the Music Accompaniment. 11 simply outputs 11111111. While the synchSMs are concurrent, note that all but one will be waiting in a wait state at any time.