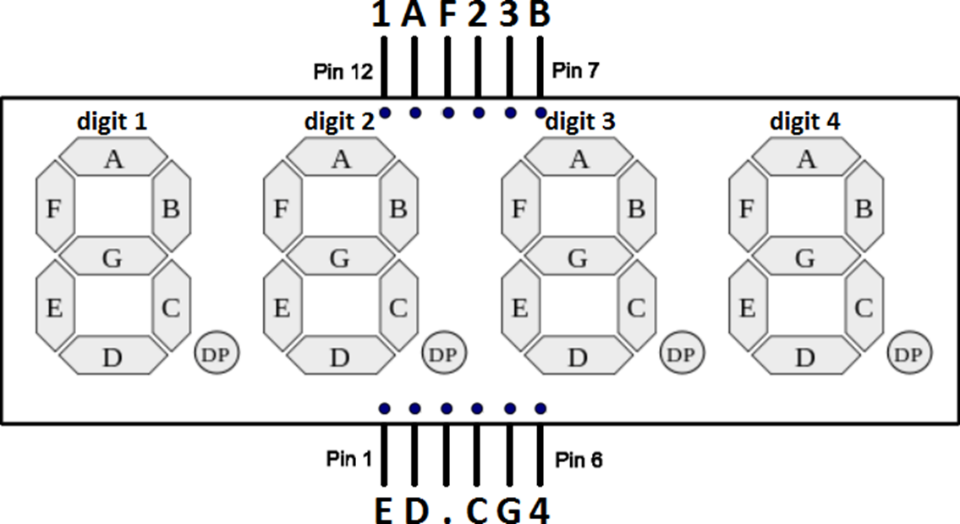
**Embedded Controllers**

**Assignment 3 – LEDs and Displays**

****

1. Watch the LinkenIn Learning video **Learning Arduino: Interfacing with Hardware** (1hr 40m) and **How a 2x16 LCD Works** (26m) from the links in Brightspace.
2. Start with a 4-digit 7-segment common anode display (3461BS-1 or equivalent). Configure a single digit to display each segment one at a time (for one second) from A to G and then display all the digits from 0 to 9.
   * This is similar to the Cylon circuit you already built except the common anode goes to 5V and all the other I/O are ground. So in this case, logic LOW turns the segment on and logic HIGH turns the segment off
   * Start by just experimenting to determine which pins are connected to which segments
   * You can also look up the data sheet to find the same information
   * Like any LED, don’t forget to include resistors to each of the cathodes to limit the current or you will burn it out
   * Start by just trying to blink 1 segment
   * Once you have each segment flashing you can add the digits 0 through 9
   * Use the array below to represent the values of each number

int num\_array[10][7] = {

{ 1,1,1,1,1,1,0 }, // 0

{ 0,1,1,0,0,0,0 }, // 1

{ 1,1,0,1,1,0,1 }, // 2

{ 1,1,1,1,0,0,1 }, // 3

{ 0,1,1,0,0,1,1 }, // 4

{ 1,0,1,1,0,1,1 }, // 5

{ 1,0,1,1,1,1,1 }, // 6

{ 1,1,1,0,0,0,0 }, // 7

{ 1,1,1,1,1,1,1 }, // 8

{ 1,1,1,0,0,1,1 }}; // 9

* Do not use the SevSeg library for this project.
* When you have it working, upload a video and include a link in the assignment along with your code. **(2 pts)**

1. Next use an SN74HC595N shift register IC to run a 60 second timer using fewer I/O pins

* Use the ICs labeled SN74HC595N and research this IC on the Internet to find the datasheet
* It’s probably easier to get one 7-segment display working first before adding the next one
* You might find this example useful <https://create.arduino.cc/projecthub/meljr/7-segment-led-displays-102-using-a-shift-register-6b6976> In it they display hex numbers on a single digit display
* You can switch between the first and second digit by controlling the 2 common anodes with I/O pins set HI rather than the +5v source
* This will involve a couple of nested loops
* Your timer should count from 0 to 59 before automatically starting over
* When you have it working, upload a video and include a link in the assignment along with your code. **(2 pts)**

1. Attach a 1602 LCD module and 2 buttons (or just the Keypad shield) and load the LiquidCrystal / HelloWorld sketch. Once it’s working modify the sketch to create a coin counter from a vending machine.
   1. If you’re using the LCD Keypad Shield, you’ll need to research to how the attached buttons work. Google the LCD KeyPad Shield and copy tutorial example 1 by Mark Bramwell. This example will display a value for each button that is pressed.
   2. Also, if you’re using the keypad shield you can still used the built in <LiquidCrystal.h> library but you modify the I/O pins to (8, 9, 4, 5, 6, 7) instead of (12, 11, 5, 4, 3, 2)
   3. The counter should have two buttons representing quarters and loonies
   4. The LCD should display the current total in $0.00 as you press the buttons
   5. Don’t forget about key bounce
   6. When you have it working, upload a video and include a link in the assignment along with your code. **(2 pts)**
2. Create a simple game on the LCD Keypad Shield.

* Enter the following program to make a custom animated character:

#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

byte armsDown[8] = {

0b00100,

0b01010,

0b00100,

0b00100,

0b01110,

0b10101,

0b00100,

0b01010

};

byte armsUp[8] = {

0b00100,

0b01010,

0b00100,

0b10101,

0b01110,

0b00100,

0b00100,

0b01010

};

void setup() {

lcd.begin(16, 2);

lcd.createChar(1, armsDown);

lcd.createChar(2, armsUp);

}

void loop() {

lcd.setCursor(7, 1);

lcd.write(1);

delay(1000);

lcd.setCursor(7, 1);

lcd.write(2);

delay(1000);

}

* + Modify the code to create a different animated character
  + Note how the arrays define the character and how lcd.setCursor() defines the position
  + Modify sketch so that one button moves your character left and another moves it right
  + Feel free to make your game more exciting
  + When you have it working, upload a video and include a link in the assignment along with your code. **(2 pts)**

1. Finally, **choose one** of the options below:
   1. Use the 7-segment TTL decoder IC (74LS48) and the common **cathode** 7-segment display to build a 0-9 display that only uses 4 data lines
      * Attach a rotary encoder so the value of the number increments clockwise and decrements counterclockwise
      * This example might be useful although it’s not using the decoder <https://simple-circuit.com/arduino-rotary-encoder-7-segment-display/>
   2. Use a TFT LCD Touch Shield to display your name.
      * There are a number of LCD shields to choose from
      * Information and libraries most shields can be found on the manufacture’s web site
      * Most libraries come with example code
      * Also remember to look for other projects where people have used the same hardware
   3. Use a 1602 LCD to create a front panel display for your computer case that displays important information such as CPU temperature or RAM usage.
   * You can use the LCD Smartie program to capture the computer info
     + <http://lcdsmartie.sourceforge.net/>
   1. Use the membrane keypad and a 1602 LCD to create a security system that will turn on a green LED when you enter the correct 4-digit code and a red LED when the wrong code is used
   * You might find this link useful <https://www.circuitbasics.com/how-to-set-up-a-keypad-on-an-arduino/>
2. When you have one of them working, upload a video and include a link in the assignment along with your code. **(2pts)**
3. If you haven’t already don’t so, remember to write the weekly quiz on Brightspace