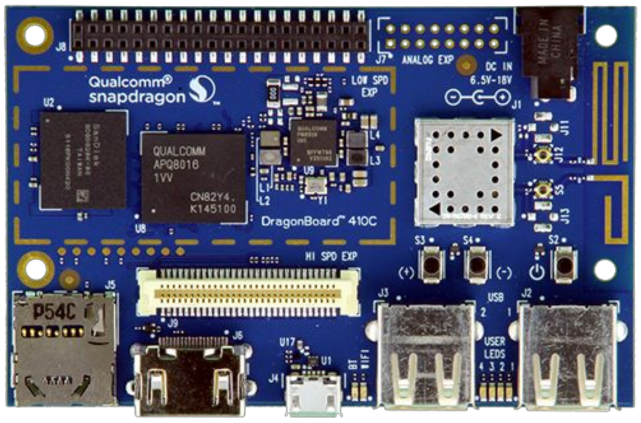


DragonBoard™ 410c

Module 5

LED Block

Table of Contents

LED Block

In this module, we will be using the DragonBoard™ 410c to interact with an LED block (or multiple LED blocks). This project will help

---

Name: LED Block

Project Description: Users will learn how to use the provided programs and libraries as a way to familiarize themselves further with programming as well as how to use an LED block for their own projects. Users should be able to have an idea of how to build their own LED block circuits and how to interact with them at the end of the course.

Project Difficulty:
Estimated Time: hour/s
HW Difficulty: */10
SW Difficulty: */10

Code Access:
The code for this module is found at the following link:
https://github.com/IOT-410c/IOT-DB410c-Course-3/tree/master/Modules/Module_5_LED_Block

To obtain this code, issue the following command:
git clone `https://github.com/IOT-410c/IOT-DB410c-Course-3`

Then inside of Android Studios, import the project found at
IOT-DB410c-Course-3/Modules/**Module_5_LED_Block**

Note:
- Ensure the DragonBoard™ 410c is connected to a display
- Ensure the correct power source is connected to the device (DC 12V, 2A)
    - **WARNING**: Exceeding the recommended power could damage the device
- Ensure the device has USB debugging enabled (Android users only)
- Ensure the DragonBoard™ 410c is connected to a computer

Parts List:

a. LED Block set
b. Wires
c. Breadboard
d. Amplifier from Module 3

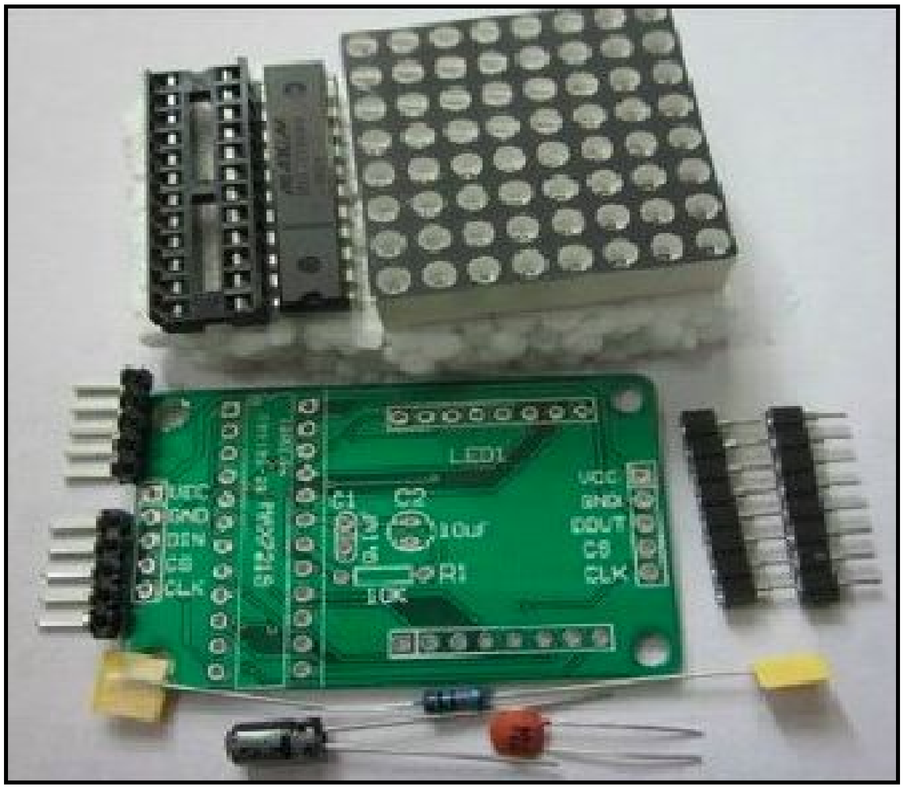
1 - LED Components

Instead of making a custom LED matrix and IC, we will be using a pre-built set (inexpensive). This section will go over the details pertaining to the components of the kit.

1.1 - LED Matrix Kit

The LED matrix kit should contain:

- an 8x8 LED matrix block
- IC and an IC socket
- male headers
- passive components (resistors, capacitors, inductors)
- PCB for the matrix and IC (requires soldering)



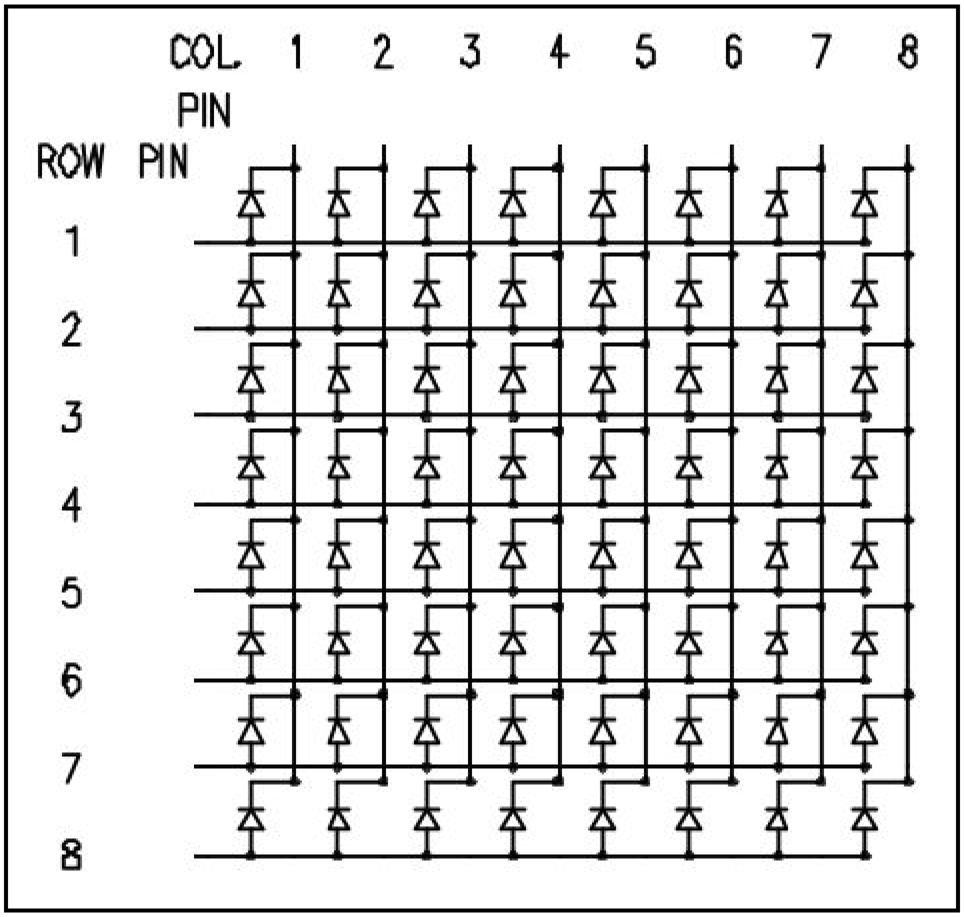
To set up the LED (component placement is printed on the PCB, shown above):

- the headers will go on the left and right sides
- the socket for the IC will go up and down in the long strip on the left side
- the resistor goes into the horizontal holes labeled R1 (any direction)
- the orange, ceramic capacitor into the C1 holes (any direction)

- the black, larger capacitor goes into the C2 spot (**Make sure you have the negative side (indicated with the white stripe, and shorter leg) into the hole marked (-) and the positive side (longer leg) into the (+)**)
- the sockets for the matrix go into the two horizontal lines on the top and bottom of the PCB
- finally, you can attach the matrix and IC into their respective sockets

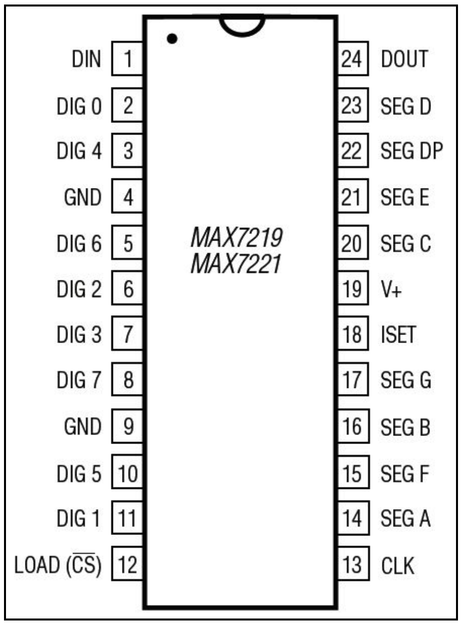
1.2 - 8x8 LED Matrix

Here is what the LED matrix looks "under the hood":



This design simply allows us to have more control over which diodes are being powered and which ones are not. And so how is that possible? Well, it's possible thanks to the IC that came with this LED matrix kit (which will be covered next)

1.3 - MAX7219 IC



The IC in this kit is called the MAX7219 (shown on the right). There isn't much need to worry about this schematic because the PCB already handles the powering and grounding of this chip; you just need to worry about putting the correct inputs into the header on your complete circuit.

Below is what is important for the programming of the LED matrix. There are 16 pins on back of the LED matrix, meaning that it can receive a 16 bit signal (each pin can be on or off). The general format of an instruction for the LED display is:

- first 4 [15:12] significant bits don't matter: can be set to 0 or 1
- the next 4 [11:8] will choose where the send the data to
- the last 8 [7:0] will select the operation to be performed or data to send (what you are trying to get the matrix to do)

For more information:
https://www.sparkfun.com/datasheets/Components/General/COM-09622-MAX7219-MAX7221.pdf

REGISTER	ADDRESS					HEX CODE
	D15–D12	D11	D10	D9	D8	
No-Op	X	0	0	0	0	0xX0
Digit 0	X	0	0	0	1	0xX1
Digit 1	X	0	0	1	0	0xX2
Digit 2	X	0	0	1	1	0xX3
Digit 3	X	0	1	0	0	0xX4
Digit 4	X	0	1	0	1	0xX5
Digit 5	X	0	1	1	0	0xX6
Digit 6	X	0	1	1	1	0xX7
Digit 7	X	1	0	0	0	0xX8
Decode Mode	X	1	0	0	1	0xX9
Intensity	X	1	0	1	0	0xXA
Scan Limit	X	1	0	1	1	0xXB
Shutdown	X	1	1	0	0	0xXC
Display Test	X	1	1	1	1	0xXF

2 - LED Matrix

This lesson will go into more detail regarding how the LED matrix works and most of the cool things you could do with this kit.

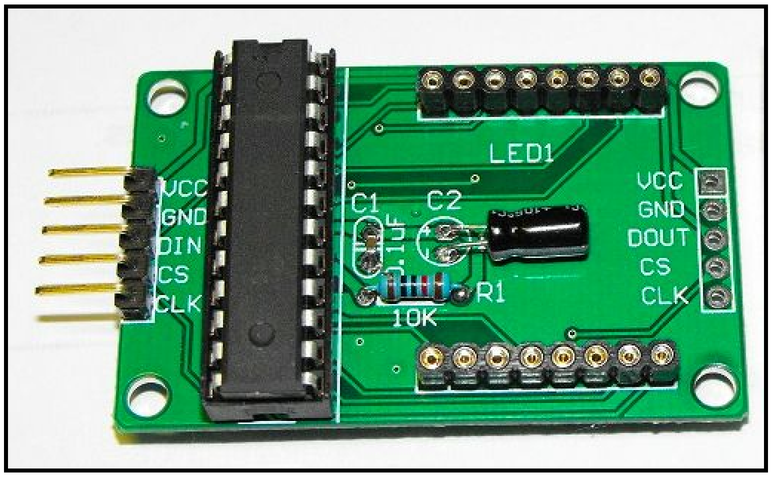
2.1 - How It Works

Using the 16 bit instructions we talked about earlier, the IC will shift those into the LED matrix which will then do its magic and execute the what you had told the system to do (easier to see this in the code).

Let’s do a quick example. Say your LED block is in shutdown mode (power savings; the LED won’t turn on if you try to power it), but now you want your lights to turn on. Using the 16 bit instruction format, how would you do this? To begin, you know that the first 4 bits are don’t cares (we will use 0’s in this case). Next, we need to find the 4 bits corresponding to where we want to send the instruction, in this case 1100 (shutdown register). And finally, we will shift in a ‘0’ (for low) to turn off shutdown mode (turn on the LED block). Therefore, our 16 bit instruction is: 0000 1100 000 0001 (or 0x0C01) to activate the matrix.

So now that we have the instruction, how does the chip receive it? To transfer instructions to the chip, we must involve the header pins (clock, load, data). Basically, we will use the clock and load to shift in data bits. Don’t worry if this doesn’t make sense; it will be easier to understand once you begin to look at the code.

2.2 - Matrix Customization



Some cool customization things you can do:

- additional blocks (connect in a line, up to 8)
    - shown on the right, if we connect another LED block to the right side (DOUT of this block goes to DIN of the next block, etc), we can have a chain of LED blocks to display something of your choice!

- Set scan limit
  - Setting the scan limit (from 0 to 7) allows you to choose how many columns will be powered in the block (i.e. saving power in case you don't need to turn on parts of the LED block)
- LED intensity
  - Setting the LED's intensity (from 0 to 15, highest = 15) will determine how bright the LEDs will shine. (i.e. saving power by setting LED intensity to a lower setting, lowering light settings because you're working at night and the LEDs are blinding you)
- Shutdown mode
  - From earlier, the shutdown mode will power off the LEDs and thus we are unable to send any signals into the matrix until this mode is turned off
- Display test
  - turns on all your LEDs (yes, all 64, be careful if your LED intensity is 15). This is mostly used to make sure none of the LEDs are burned out

3 - LED Block Set Up

This lesson will be covering the libraries that we wrote (code heavy section). In here, you should get a better picture of how the things that were explained above are used.

3.1 - LED Processor Library

We highly recommend you watch the videos but here is a quick run through of the Python code:

- **Constructor** (__init__(self, devices=1, intensity=5, scanlimit=7))
  - We already preset the constructor values for you (feel free to change based on your needs).
  - Devies = # of devices that are being used (default 1)
  - Intensity = brightness of the lights (default 5)
  - ScanLimit = # of digits to display (default 7; all of them)
  - The constructor will make sure all the LED blocks (# of devices) will be on (shutdown=0) and set initiate their intensities and scanlimit
- **getStatus**
  - Returns the status of a field
- **createInstruction**
  - sets up the instruction with desired address and signal
- **transfarInstruction** (try to understand this)
  - allows us to set the instruction we created in createInstruction to the LED block
- **setRow, setColumn, setLED**
  - these will set an entire row or column to on or off as well as set an individual LED on or off
- **printLetter, printNumber, printDice, printWord**
  - these are mostly used for demoing -- lets you print out something for the LED to display
- **clearDisplay, clearDisplays**
  - clears specified display or all of them
- **setIntensity, setScanLimit, shutdown**
  - allows user to change (set) these states to a desired mode
- **displayTest, exitTest**
  - begin a display test or end it
- **cleanup**
  - the deconstructor

The rest of the functions are used for demos! Feel free to test them out on your own system to make sure the LEDs work as well as help you understand how the code works better (if something looks wrong or is unoptimized, feel free to modify the code and make it better!)

3.2 - Character Library

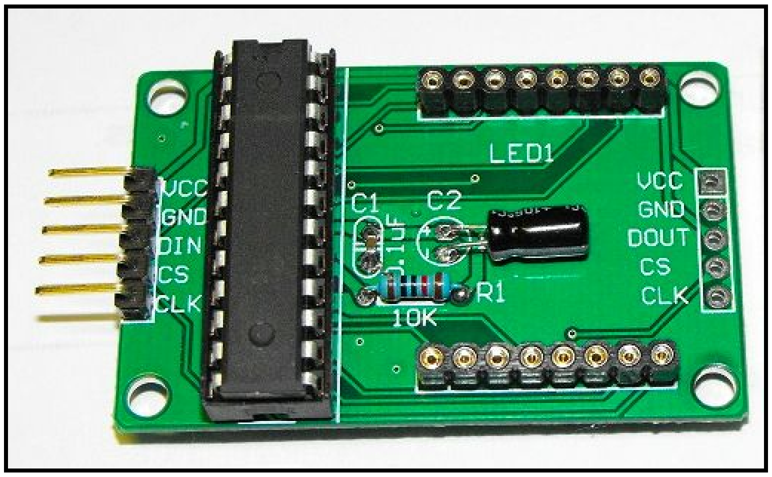
This section go over now the LED block is able to print out letters: the code actually uses this character library which contains most letters and turns on the corresponding LEDs to make a display of the letters. (**Note**: All the 0B's in front of the numbers indicate that the following number is a binary string) Let's go over the functions really quick:

- **setLetter, setLetterR**
    - sets the LED to display a 5x5 sized letter on the display; setLetterR will do the same thing but rotated 90° (for putting a block on it's side)
- **setNumber**
    - similar to setLetter but for numbers
- **setDiceNumber**
    - displays a face of a dice (1, 2, 3, 4, 5, 6)

Once again, feel free to add new and cool functions, add new characters to print out, add lower case characters or even change what a character looks like!

3.3 - Expansion to Multiple Blocks

Remember, to expand the LED block, you will need to attach your second (or next LED block) block's left header (VCC, GND, DIN, CS, CLK) to the previous block's right header (VCC, GND, DOUT, CS, CLK).



Next, make sure in your code that you specify how many LED blocks you are using (stored as a variable called devices or numOfDevices) and the rest of the methods will properly modify the blocks for you to have a proper/desired display.

4 - Build and Demo

Now that we have gone over the components and of the LED block kit as well as the libraries we will be using, this lesson will go over the build of the circuit (powering and connecting to the DragonBoard™ 410c). And after building the circuit, let's implement it with some programs

4.1 - Building the Circuit

First, make sure your amplifier is set up properly (module 3) and is powered and grounded. In these demos, we will be using pins 23, 24, 25 (GPIOs) to power the LED.

Now, let's connect the LED block to the amplifiers (if you haven't, take the time to build the LED block). The amplifier taking in input from pin 23 should output to DIN (data in), pin 24 should be connected to CS (the load), and pin 25 should be connected to CLK (clock). Finally, power and ground the LED block (VCC and GND). **Note:** You can always change which pins are used in your circuit so long as you make sure make the appropriate changes in the code.

With that all complete, let's run some demos!

4.2 - Message Display (Ubuntu)

After pulling the code, make sure you set the code to have the correct number of devices you will be using (in this demo, we use 2 devices). Now, you can just simply run the program and write a message in that will be displayed on the LED blocks.

To test your mastery, try to change the delays or maybe even make the message you type in flash (try adjusting light intensity) at times.

4.3 - Dice Game (Ubuntu)

Don't have 1-8 die but you have a DragonBoard™ 410c, some circuits, and 1-8 LED blocks? No problem! Load up the program we have for you, and starting rolling the die!

This one is also really simple; make sure you have the correct number of devices in the code (we have 2) and run it. This will randomly generate some numbers 1-6 and then it'll display those numbers onto the LED.

As a challenge, try to increase the number dice faces (this would be made in the character library, don't forget to reflect the change in the dice game code).

4.4 - Calit Bird Game (Android)

Pull code from Git links provided at the beginning of this document. Follow steps from Course 2 in order to get application pushed to board.

References

LED Matrix kit set up:
http://tronixlabs.com/news/getting-started-with-your-8x8-red-led-matrix-serial-display-module-kit-tronixlabs-australia/

LED IC Chip information:
https://www.sparkfun.com/datasheets/Components/General/COM-09622-MAX7219-MAX7221.pdf