CSC270—Circuits & Systems

Week 7 — Spring 2019

Dominique Thiébaut
dthiebaut@smith.edu
• If a question asks whether something is a latch or not, the only valid answers are "Yes, it is a latch," and "No, it is not latch," followed by "because…"

• If a signal is **active-low**, put a circle on its input or output of a circuit, and put a bar over it.

• If you participate on a homework with somebody else, please list that person's name. Otherwise…

• when presented with a Karnaugh map, try to solve it first to see what the function simplifies to. This may give you alternatives for implementing it with a mux.
• You cannot use an oscilloscope to measure a resistor. Scopes are good for measuring high-frequency signals.

• In the GYR FSM, if Green and Red are never on at the same time, then it's simpler to make

\[ R = G' \]
Comments on Lab Report

• Make sure you simulate all the transient states, otherwise you are NOT fully simulating your FSM…

```python
# simulate the FSM staring in various states...
for Q2,Q1,Q0 in [(1,1,1), (1,1,0), (1,0,1)]:
    print( "\nStaring with Q2, Q1, Q0 = ", Q2, Q1, Q0 )
    for steps in range( 20 ):
        ...
```

• Set DPI to 300 when scanning diagrams for your reports
The Arduino
History

- The Arduino project was started at the Interaction Design Institute Ivrea (IDII) in Ivrea, Italy. In 2003 Hernando Barragán created the development platform Wiring as a Master's thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas.

https://en.wikipedia.org/wiki/Arduino
• In 2001, together with MIT PhD candidate Ben Fry, Reas created the Processing programming language.[16] Processing is widely used by thousands of artists and designers worldwide, and by educators teaching the fundamentals of programming in art and design schools.[17][18]
```c
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);

  // the loop function runs over and over again forever
  void loop() {
    digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
    delay(1000);                       // wait for a second
    digitalWrite(LED_BUILTIN, LOW);   // turn the LED off by making the voltage low
    delay(1000);                      // wait for a second
  }
}
```
The Mega2650 Arduino (Made by ELEGOO)

"Established in 2011, Elegoo Inc. is a thriving technology company dedicated to open-source hardware research & development, production and marketing. Located in Shenzhen, the Silicon Valley of China, we have grown to over 100+ employees with a 10,763+ square ft. factory."

https://www.elegoo.com/
References

- https://www.arduino.cc/
Outline

- Introduction
- Physical Layout
- Specifications
- Power
- Memory
- Programming
  - Digital I/O
  - Analog I/O
Arduino MEGA Pinout

www.CircuitsToday.com
4-min Intro

https://www.youtube.com/watch?v=3Ek7UEPbkqg
TTL = Transistor-Transistor Logic = 5V

Arduino Mega 2560 Pinout

Pinout
Mega: Specs

- Controller based on **Atmega2560**
- **8-bit** microcontroller. RISC architecture
- **54 digital** IO pins (15 can be used in PWM)
- **16 analog** pins
- **16MHz** crystal
- Programmed with custom **IDE**
- Serial communication ports
Mega: Specs (cont'd)

- Operating voltage: **5V**
- Power supply: 7-12V
- Flash memory: 256KB
- RAM 8KB
- EEPROM 4KB
**USB.** Connect to a laptop via a USB Cable.

**VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

**5V.** This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

**3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

**GND.** Ground pins.
Power

- +5V
- +7/12V
- 3.3V, 50mA
- 5V
- 6-12V
How **Long** can a **9V** Battery Power the **Arduino**?

- Hard to tell: depends on what's connected

- **Solution**: make Arduino monitor itself (with analog input)!

• **256 KB of flash memory** where the sketch is stored (of which 8 KB is used for the bootloader).

• **8 KB of SRAM** where the sketch creates and manipulates variables when it runs (strings of chars are stored there).

• **4 KB of EEPROM** which is memory space that programmers can use to store long-term information.

• Flash memory and EEPROM memory are **non-volatile** (the information persists after the power is turned off). SRAM is volatile and will be lost when the power is cycled.
Memory (cont'd)

• If your sketch talks to a program running on a (desktop/laptop) computer, you can try shifting data or calculations to the computer, reducing the load on the Arduino.

• If you have lookup tables or other large arrays, use the smallest data type necessary to store the values you need.

• If you don't need to modify the strings or data while your sketch is running, you can store them in flash (program) memory instead of SRAM; to do this, use the PROGMEM keyword.

https://www.arduino.cc/en/tutorial/memory
Outline

- Programming
  - Pins
    - Digital I/O
    - Analog Input
    - "Analog" Output
    - Serial Communications
Programming the Mega2650

Here is an example of code that can be used to program the Mega2650 using Arduino IDE:

```c
void setup() {
  // Initialize the digital pin as output.
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH);  // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);   // set the LED off
  delay(1000);             // wait for a second
}
```
Blink

Turns on an LED on for one second, then off for one second, repeatedly.

This example code is in the public domain.

*/

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
 void setup() {
   // initialize the digital pin as an output.
   // Pin 13 has an LED connected on most Arduino boards:
   pinMode(13, OUTPUT);
 }

 void loop() {
   digitalWrite(13, HIGH); // set the LED on
   delay(1000); // wait for a second
   digitalWrite(13, LOW); // set the LED off
   delay(1000); // wait for a second
 }

 Typical Behavior

 Initialization  Check sensors, control devices
Programming: Two Options

setup

setup loop loop loop loop loop

on the loop body empty
Programming: Two Options

```c
void loop() {
    // --- compute ---
    count++; // count is incremented
    x = func( count ); // function call
    // --- wait 1 sec ---
    delay( 1000 ); // delay for 1 second
}
```

- `int count;` is incremented in the `loop` function.
- There is a call to a function `func`.
- A `delay` function is called to wait 1 second.

This diagram illustrates the process:

1. **Setup**: Initializes the environment.
2. **Loop**: Performs computations and function calls.
3. **Delay**: Introduces delays to simulate time.
4. **Time Progression**: Shows the progression of time with delays.
Important

• When the **arduino** is powered back up, it automatically restarts the last sketch that was uploaded to it:
  
  • **setup()** is executed again
  
  • **loop()** is repeated forever
Printing is different...
Serial.print()

Serial Monitor (what it is used to mean)

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println("x = ");
  Serial.println(x);
}
```

Band = bps = bits per second

x = 3

Serial Monitor

Hello!
Hello!
Hello!
Hello!
Hello!
Hello!
Hello!
Hello!
Hello!
Hello!

Done uploading.

Sketch uses 1950 bytes (0%) of program storage space. Maximum is 2539.
Global variables use 194 bytes (2%) of dynamic memory, leaving 7998 b
Serial.print()
Challenge: Explain This!

very likely pruned by laptop

15:59:44.030 → Start!
15:59:44.068 →
15:59:49.050 →
15:59:54.030 →
15:59:59.042 →
16:00:04.027 →

??
Challenge: Explain This!

Some questions about previous slide

- Does the timing make sense?
- What does 9600 mean?
- Do we see the 5-second delay?
- "Who" prints the time we see in the Serial Monitor window?
- "Who" computes these time markers?
Blink!
Exercise

- Go to https://www.arduino.cc/en/Guide/ArduinoMega2560
- Setup your Arduino, and run the **Blink sketch** on your AtMega2560
Challenge: How Fast is The ELEGOO Mega2650?
Solving the N-Queens Problem

https://www.youtube.com/watch?v=ckC2hFdLff0
C Program for Arduino

N-Queens on the Arduino

D. Thiebaut (talk) 10:17, 27 February 2019 (EST)

- Make sure you set the baud rate to 9600 in your IDE.
- You may also want to turn on the "Show timestamp" option on the Serial Monitor

Source [edit]

/*
   queensdemo.c
D. Thiebaut
Position N queens on an NxN chess board

   Typical output:
   0 2 4 1 12 8 13 11 14 5 15 6 3 10 7 9

http://www.science.smith.edu/dftwiki/index.php/N-Queens_on_the_Arduino
Side Note: Importing C code Into Arduino

```c
#include <stdio.h>

int main( int argc, char *argv[] ) {
    printf( "Hello World!\n" );
    return 0;
}
```

```c
int main2( int argc, char *argv[] ) {
    //printf( "Hello World!\n" );
    Serial.print( "Hello World!\n" );
    return 0;
}
void setup() {
    Serial.begin( 9600 );
    main2( 0, NULL );
}
void loop() {
    // put your main code here, to run repeatedly:
}
```
### Benchmarks

(All times in ms)

<table>
<thead>
<tr>
<th>Board size</th>
<th>Macbook Pro 2009</th>
<th>Macbook Pro 2014</th>
<th>MacPro 2009</th>
<th>MacPro 2014</th>
<th>Linux Mint Beowulf2</th>
<th>Laptop 1</th>
<th>Linux 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>8x8</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9x9</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10x10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11x11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12x12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13x13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14x14</td>
<td>26</td>
<td>3</td>
<td>8</td>
<td>5</td>
<td>17</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>15x15</td>
<td>17</td>
<td>2</td>
<td>5</td>
<td>2</td>
<td>14</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>16x16</td>
<td>37</td>
<td>14</td>
<td>27</td>
<td>18</td>
<td>46</td>
<td>1</td>
<td>33</td>
</tr>
<tr>
<td>17x17</td>
<td>28</td>
<td>10</td>
<td>19</td>
<td>13</td>
<td>31</td>
<td>1</td>
<td>60</td>
</tr>
<tr>
<td>18x18</td>
<td>87</td>
<td>34</td>
<td>67</td>
<td>41</td>
<td>104</td>
<td>5</td>
<td>16</td>
</tr>
<tr>
<td>19x19</td>
<td>19</td>
<td>7</td>
<td>13</td>
<td>9</td>
<td>22</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td>20x20</td>
<td>144</td>
<td>64</td>
<td>119</td>
<td>70</td>
<td>152</td>
<td>24</td>
<td>99</td>
</tr>
<tr>
<td>21x21</td>
<td>36</td>
<td>13</td>
<td>26</td>
<td>17</td>
<td>50</td>
<td>1</td>
<td>29</td>
</tr>
<tr>
<td>22x22</td>
<td>750</td>
<td>311</td>
<td>662</td>
<td>384</td>
<td>610</td>
<td>221</td>
<td>463</td>
</tr>
<tr>
<td>23x23</td>
<td>81</td>
<td>31</td>
<td>63</td>
<td>39</td>
<td>101</td>
<td>3</td>
<td>58</td>
</tr>
<tr>
<td>24x24</td>
<td>242</td>
<td>101</td>
<td>199</td>
<td>119</td>
<td>226</td>
<td>56</td>
<td>157</td>
</tr>
<tr>
<td>25x25</td>
<td>92</td>
<td>37</td>
<td>72</td>
<td>44</td>
<td>111</td>
<td>7</td>
<td>64</td>
</tr>
<tr>
<td>26x26</td>
<td>248</td>
<td>102</td>
<td>201</td>
<td>121</td>
<td>227</td>
<td>58</td>
<td>161</td>
</tr>
<tr>
<td>27x27</td>
<td>284</td>
<td>118</td>
<td>232</td>
<td>142</td>
<td>257</td>
<td>69</td>
<td>184</td>
</tr>
<tr>
<td>28x28</td>
<td>1551</td>
<td>633</td>
<td>1250</td>
<td>734</td>
<td>1213</td>
<td>469</td>
<td>938</td>
</tr>
<tr>
<td>29x29</td>
<td>1518</td>
<td>356</td>
<td>687</td>
<td>413</td>
<td>684</td>
<td>244</td>
<td>522</td>
</tr>
<tr>
<td>30x30</td>
<td>33667</td>
<td>12586</td>
<td>23558</td>
<td>14158</td>
<td>22889</td>
<td>9041</td>
<td>17252</td>
</tr>
</tbody>
</table>

- **Macbook Pro 2009**: 453 ms
- **Macbook Pro 2014**: 3529 ms
- **MacPro 2009**: 238 ms
- **MacPro 2014**: 18557 ms
- **Linux Mint Beowulf2**: 860 ms
- **Laptop 1**: 172089 ms
- **Linux 8**: 440000 ms
Outline

• Programming

• Pins

• Digital I/O
  • Analog Input
  • "Analog" Output
  • Serial Communications