• 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

• 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/Ardueño
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]
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 level 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/

• http://www.science.smith.edu/dftwiki/index.php/CSC270_Weekly_Schedule_2019
Outline

• Introduction
• Physical Layout
• Specifications
• Power
• Memory
• Programming
  • Digital I/O
  • Analog I/O
4-min Intro

https://www.youtube.com/watch?v=3Ek7UEPbkqg
TTL = Transistor-transistor logic = 5V
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)!

Memory

• **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

```c
void setup()
{
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, HIGH);
}

void loop()
{
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
}
```
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**
- **loop**
- **loop**
- **loop**
- **loop**
- **loop**

Note: The loop body is empty.
Programming: Two Options

void loop() {
  // --- compute ---
  count++;
  x = func(count);
  // --- wait 1 sec ---
  delay(1000);
}

setup

```
SerialLoop | Arduino 1.8.8
```

loop

delay

loop

delay

loop

delay

```
Sketch uses 1950 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 194 bytes (2%) of dynamic memory, leaving 7998 bytes free.

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/cu.usbmodem1411
```
• 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 trying to mean)

```
Serial.begin(9600);
void loop() {
    Serial.println("Hello!");
    delay(1000);
}
```

Band = bps = bits per second

```
\{ Serial.print("x = ");
    Serial.println(x);
\}
```

Serial Monitor

\(x = 3\)
Serial.print()
Challenge: Explain This!

very likely printed by laptop

D. Thiebaut, Computer Science, Smith College
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;
}
```

```c
void setup() {
    Serial.begin(9600);
    main2(0, NULL);
}
```

```c
void loop() {
    // put your main code here, to run repeatedly:
}
```

Done uploading.

Sketch uses 1/92 bytes (0%) of program storage space. Maximum is 2539
Global variables use 198 bytes (2%) of dynamic memory, leaving 7994 b

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on /dev/cu.usbmodem1411
## 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>1</td>
<td>1</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>1</td>
<td>1</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>99</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>29</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>
Outline

• Programming

• Pins

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

- 54 digital pins
- `pinMode( pin_number, mode )`
- `digitalRead( pin_number )`
- `digitalWrite( pin_number, value )`

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

void loop() {
  digitalWrite(13, HIGH);  // sets the digital pin 13 on
  delay(1000);  // waits for a second
  digitalWrite(13, LOW);  // sets the digital pin 13 off
  delay(1000);  // waits for a second
}
```
**Input and Output**

Each of the 54 digital pins on the Arduino 2560 Mega can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20–50 kOhms.

In addition, some pins have specialized functions:

- **Serial:** 0 (RX) and 1 (TX);
- **Serial 1:** 19 (RX) and 18 (TX);
- **Serial 2:** 17 (RX) and 16 (TX);
- **Serial 3:** 15 (RX) and 14 (TX).

Digital IO

Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega16U2 USB-to-TTL Serial chip.

**External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

**PWM:** 2 to 13 and 44 to 46. Provide 8-bit PWM output with the analogWrite() function.

**SPI:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

**LED:** 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it’s off.

**TWI:** 20 (SDA) and 21 (SCL). Support TWI communication using the Wire library. Note that these pins are not in the same location as the TWI pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and analogReference() function.

There are a couple of other pins on the board:

https://www.electroschematics.com/7963/arduino-mega-2560-pinout/
Digital Output

What value of R should we use?
How many LEDs can we drive from one IO pin?
```cpp
void setup() {
  pinMode(13, OUTPUT); // sets the digital pin 13 as output
}

void loop() {
  digitalWrite(13, HIGH); // sets the digital pin 13 on
  delay(1000); // waits for a second
  digitalWrite(13, LOW); // sets the digital pin 13 off
  delay(1000); // waits for a second
}
```
Digital Input
int ledPin = 13;  // LED connected to digital pin 13
int inPin = 7;    // pushbutton connected to digital pin 7
int val = 0;      // variable to store the read value

void setup() {
    pinMode(ledPin, OUTPUT);  // sets the digital pin 13 as output
    pinMode(inPin, INPUT);    // sets the digital pin 7 as input
}

void loop() {
    val = digitalRead(inPin);  // read the input pin
    digitalWrite(ledPin, val); // sets the LED to the button's value
}
(Simplest) Digital Input

on-board resistor
(Simplest) Digital Input

```cpp
void setup() {
  // start serial connection
  Serial.begin(9600);

  // configure pin 2 as an input and enable the internal pull-up resistor
  pinMode(2, INPUT_PULLUP);
  pinMode(13, OUTPUT);
}

void loop() {
  // read the pushbutton value into a variable
  int sensorVal = digitalRead(2);

  if (sensorVal == HIGH) {
    digitalWrite(13, LOW);
  } else {
    digitalWrite(13, HIGH);
  }
}
```
Analog Input

- 10-bit analog to digital converter
- Pins A0 to A14
Analog Input

- 10-bit analog to digital converter
- Pins A0 to A14

```c
int analogPin = A3; // potentiometer wiper (middle terminal) connected to analog pin 3
                  // outside leads to ground and +5V
int val = 0;      // variable to store the value read

void setup() {
    Serial.begin(9600);  // setup serial
}

void loop() {
    val = analogRead(analogPin);  // read the input pin
    Serial.println(val);           // debug value
}
```
Analog Input

- 10-bit analog to digital converter
- Thermistor
- Potentiometer
- Joystick
- Pressure sensor
- Light sensor
- Battery (check nominal voltage)
Analog Input

- 10-bit analog to digital converter
- Thermistor
- Potentiometer
- Joystick
- Pressure sensor
- Light sensor
- Battery (check nominal voltage)
Analog Output

- "Fake" analog output
- Pulse-Width Modulation
- Pins 2 - 13 and 44 - 46
Analog Output

```c
int ledPin = 9; // LED connected to digital pin 9
int analogPin = 3; // potentiometer connected to analog pin 3
int val = 0; // variable to store the read value

void setup() {
  pinMode(ledPin, OUTPUT); // sets the pin as output
}

void loop() {
  val = analogRead(analogPin); // read the input pin
  analogWrite(ledPin, val / 4); // analogWrite values from 0 to 255
  // analogRead values go from 0 to 1023
}
```

No need to call pinMode()
Challenge

What is the frequency used by the Arduino to modulate signals sent in PWM mode?
Ready for Lab 7!