VIRTUAL COURSE
BUILD YOUR OWN DATA LOGGER

WILDLABS.NET
[ The conservation technology network ]

FREAKLABS
MODULE 3:
PROGRAMMING OUR BASIC FUNCTIONALITY

SUBMODULE-6

INTERRUPTS
Excuse me ... may I interrupt you?

- Interrupt our main application but we don’t know when
- Can make devices unpredictable
- Difficult to debug

BUT very useful!

- Key to our low power strategy
Sequential Programming

• Instructions executed in order they’re written
• We control logic and order
• Predictable!
Current WildLogger Code

• Is sequential programming

• Executes the code in the order it’s written

• We control the logic

1 – runs once
2 – runs once
3 - Repeats forever
A World Without Interrupts

```c
void loop()
{
    Serial.println("hello");
    delay(1000);
    Serial.println("hello2");
    delay(1000);
}
```
Asynchronous Events
Asynchronous Events

Excuse me!
Asynchronous Event
Interrupts and the Human Condition

EMAIL!

I have a question!

SOCIAL MEDIA!

KIDS!
Waiting for an asynchronous event
Notifications
Event Based Programming

Event occurs

Motion Detected!!

Stop everything that’s going on!

Triggers MCU

Got the message!

Event is dealt with, ie: flag is set

We’ll handle it from here

Process

Action is taken

Ok we’ve handled it!
Event Based Programming - Polling

```c
void loop()
{
  int motion = digitalRead(pinMotion);
  if (motion == HIGH)
  {
    Serial.println("Motion detected!");
  }
}
```
Event Based Programming - Polling

- reliable
- predictable
- safe
- fine if you’re not worried about power consumption
- can’t poll when device is in low power mode
Power Saving Strategy

99% of the time!
Power Saving Strategy

99% of the time!

Interrupt or Reset!
Wake up!!
Strategy behind low power mode
Strategy behind low power mode
How Does It Work on Hardware?

- Special pin called an interrupt pin
- Interrupt pin is sensitive to ‘transitions’ or when it changes from LOW->HIGH or HIGH->LOW
- When a transition occurs, an interrupt occurs and MCU is notified
- MCU stops everything, jumps to interrupt function (or the interrupt service routine - ISR) to handle the interrupt
- Interrupt pin needs to be initialized and interrupt function mapped
Wildlogger Interrupt Mapping

<table>
<thead>
<tr>
<th>Interrupt Source</th>
<th>Interrupt Number</th>
<th>Pin Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Time Clock</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>PIR Motion Sensor/Aux Intp</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Pushbutton 0</td>
<td>2</td>
<td>6</td>
</tr>
</tbody>
</table>
Good Tips

Rule of thumb:
• Keep ISR as short as possible
• Do the minimum operations to deal with it and then jump out quickly!
• BAD things happen if you have long interrupt service routines
• LOTS of nasty, hard to find bugs, device becomes unreliable
On WildLogger

• In this course, only using them for a simple wakeup
• Mainly just for power management in WildLogger application
• Spending very short amount of time in our ISR – only setting a flag
COMING UP
Lab 6a:
Pushbuttons and Interrupts