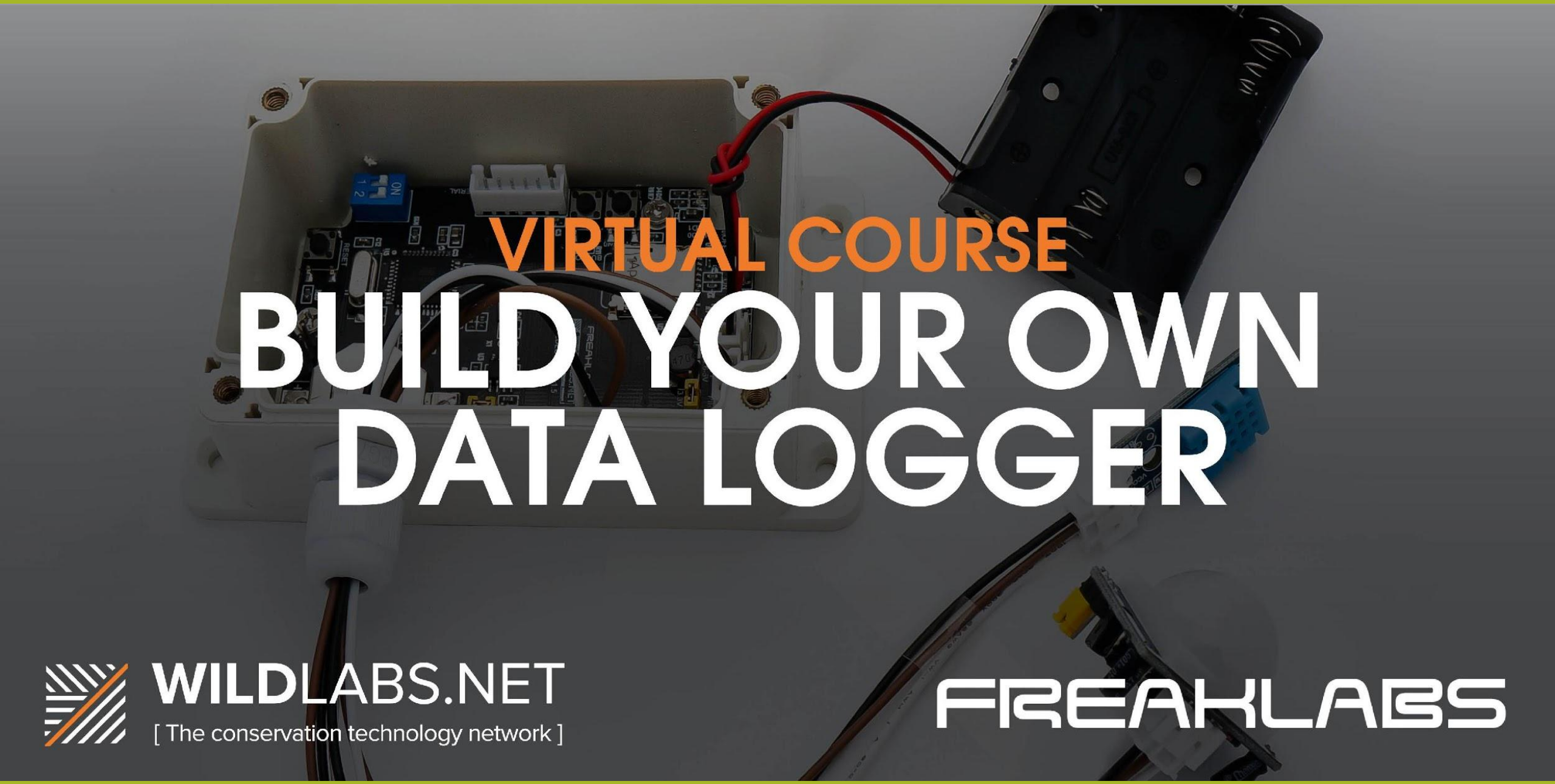

VIRTUAL COURSE
BUILD YOUR OWN
DATA LOGGER
WILDLABS.NET
[The conservation technology network]
FREAKLABS

MODULE 5-1

WHO LET THE WATCHDOGS OUT!

Watchdog Timers– A Brief Introduction

- a countdown timer that resets the microcontroller if the main program doesn't 'check-in' and reset the watchdog timer within a specified interval
- For example, every 8 seconds, the main program needs to reset the watchdog timer, or the watchdog timer will reset the system
- a watchdog timer usually lives in the main loop, or where there's long delays
- watchdog timers protects against system hangs or crashes

Types of Software Failures

- **Logic Bugs** - most common, usually occur early, usually easy to fix
- Stack or Buffer Overflow
- Memory Leaks
- Race Conditions

Types of Software Failures

- Logic Bugs
- **Stack or Buffer Overflow -** a programming error in which an attempt to write data to a particular block of memory fails because there is no space left in the block
- Memory Leaks
- Race Conditions

```
char buf[10];
for (int i=0; i<12; i++)
   buf[i] = i;
```

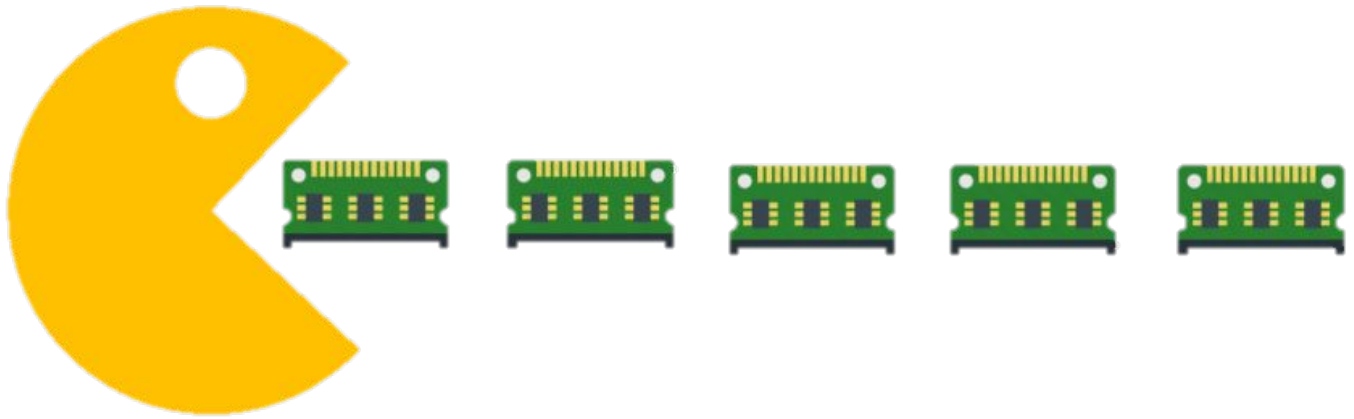
10 bytes of memory allocated here

12 bytes of memory will be used in this loop

After the 10th byte of memory is assigned, the next byte will cause the application to either crash, or make the system behave unpredictably. This is because we will be overwriting memory that's used for something else

Types of Software Failures

- Logic Bugs
- Stack or Buffer Overflow
- **Memory Leaks -** A memory leak is any portion of an application which uses RAM without eventually freeing or 'deallocating' it (for example, after execution)
- Race Conditions



Types of Software Failures

- Logic Bugs
- Stack or Buffer Overflow
- Memory Leaks
- **Race Conditions** - A race condition or race hazard is the behavior of a system where the output is dependent on the sequence or timing of other uncontrollable events. It becomes a bug when events do not happen in the order the programmer intended.

```
reg = readReg(address); // read current value from register
reg = reg | 0x01;  // set bit 0 to 1      <-- If chip register changes here, race condition bug occurs
writeReg(address, reg); // write register back
```

A chip register is a configuration location in the chip's memory accessible by the microcontroller. Bits in the register can be thought of as switches that turn on or off specific features of the chip or indicate a specific status of the chip.

Types of Software Failures

- Logic Bugs
- Stack or Buffer Overflow
- Memory Leaks
- Race Conditions

Watchdog Timers and Sleep

Goal

- Initialize the watchdog timer
- Create a command that simulates a hang event
- Observe as the system resets itself

Why Is This Important?

- Greatly improves reliability of field devices

What Do I Need to Know?

- **#include <avr/wdt.h>**
  - Need to include the watchdog timer built-in library
  - Also need special bootloader to allow for watchdog timer
- **wdt_enable(WDTO_8S);**
  - Enable the watchdog timeout to 8 seconds
  - Also:
    - WDTO_1S, WDTO_2S, WDTO_4S
- **wdt_reset();**
  - Reset the watchdog in the main loop
- **wdt_disable();**
  - Disable watchdog timer

Typing Code Goes Here

Code

Lab-5b-WatchdogTimer

```
#include <cmdArduino.h>
#include <avr/wdt.h>

int pinLed = 4;

void setup()
{
    pinMode(pinLed, OUTPUT);
    digitalWrite(pinLed, LOW);

    cmd.begin(57600);
    Serial.println("Lab 5b - Who Let the Watchdog Out?");
    Serial.println("Reset occurred.");

    cmd.add("wdtest", cmdWatchdogTest);

    wdt_enable(WDTO_8S);
}

void loop()
{
    cmd.poll();
    wdt_reset();
}
```

Code

```
25
26 void cmdWatchdogTest(int argCnt, char **args)
27 {
28     digitalWrite(pinLed, HIGH);
29     Serial.println("Testing Watchdog. About to simulate hang situation.");
30
31     // hang here
32     while (1)
33     {
34         ;
35     }
36 }
```

Show system working

COMING UP

Module 5.2: The Grand Finale