

Build Your Own Data Logger: Module 2

Terminology Starter Guide

| Video | Module 2-1 |
|------------------------------|--|
| Solid State Drive (SSD) | <p>SSD is a storage medium that uses non-volatile memory to hold and access data.</p> <p>Unlike a hard drive, an SSD has no moving parts, which gives it advantages, such as faster access time, noiseless operation, higher reliability, and lower power consumption.</p> |
| Reliability | How robust a device is against failure, especially while in the field. |
| Truck Roll | A service truck that goes out to service or replace a device if it fails in the field. This is often you, and it becomes more painful and expensive the more remotely it's located. |
| Recovery / Graceful Recovery | The device can recover and restart itself if it hangs or crashes because of a fatal error or uncaught bug. |
| Usability | <p>How easy the device and application is for people to use or service.</p> <p>Defined by the application.</p> |
| | |
| Video | 2-2 |
| Board Description Files | <p>Files used by the Arduino IDE to download code to the board, define pin numbers, and how pins map to those numbers and names.</p> <p>Created by board manufacturer.</p> |
| Board Manager | A menu item in the Arduino IDE accessed by Tools > Board > Board Manager. |

| | |
|-------------------------------------|---|
| | Used to download, install and update board files |
| | |
| Video | 2-3 |
| Sketch (or program) | <p>What a file in Arduino is called.</p> <p>The file is actually a programme that is compiled into an executable program that runs on the device.</p> |
| Signed Variable | A variable that can contain positive or negative numbers or integers as values. |
| Unsigned Variable | A variable that can contain only positive numbers or integers as values. |
| Macros | <p>Macros are special commands in C++ that are interpreted before the compiler starts.</p> <p>They are sometimes referred to as pre-compiler macros and used for special functions. We will only be using #include to include libraries in our code and #define to define constant values in this course.</p> |
| Compiler | <p>A compiler is a software program that transforms high-level source code that is written by a developer in a high-level programming language (eg. our application) into a low level object code (binary code) in machine language, which can be understood by the processor.</p> <p>The process of converting high-level programming into machine language is known as compilation.</p> |
| | |
| Video | 2-4 |
| Digital to Analogue Converter (DAC) | <p>A peripheral that changes digital values to analogue values.</p> <p>An example of a digital to analogue conversion is mapping the values 0 to 1000 to a voltage of 0 to 3.3V. In this context a value of 500 would correspond to 1.65V or half of 3.3V.</p> |

| | |
|-----------------------|--|
| | (see also Analogue to Digital converter – ADC). |
| 1000 milliseconds | 1 second |
| Duty Cycle | <p>A duty cycle is the fraction of one period in which a signal or system is active.</p> <p>Duty cycle is commonly expressed as a percentage or a ratio.</p> <p>For example if a device wakes only once per hour (3600 seconds) and stays active for 1 second before going back to sleep, the inactive duty cycle is 3599 seconds / 3600 seconds or 0.9997.</p> <p>We refer to this as a duty cycle of 0.9997 or sleeping 99.97% of the time.</p> |
| Relay | <p>An electronic relay is a type of electronic switch that opens or closes the circuit contacts based on a control signal. For example, to turn on the relay and hence the appliance connected to it, you would send a HIGH signal to its control pin. To turn off the appliance, you would send a LOW signal to its control pin.</p> <p>Relays are used wherever it is necessary to control a high power or high voltage circuit with a low voltage circuit.</p> <p>An example is controlling an 110V AC fan with the WildLogger board.</p> |
| Video | 2-5 |
| Serial Communications | <p>Sending data sequentially or 1 bit at a time over a single line or pin.</p> <p>For example for 8-bit communications, once all 8 bits have been received serially,, they're assembled into a byte, then decoded.</p> <p>Serial communication has better signal integrity, can do higher speeds, and is one of the cheapest modes of communication that can be implemented.</p> |

| | |
|--------------------------------|--|
| <p>Parallel Communications</p> | <p>Sends multiple bits at the same time, in parallel over multiple lines. Transmission and receipt of all bits occurs simultaneously.</p> <p>This used to be common in printers. They also used to be common in communicating with imaging devices like cameras. They've since been replaced by a serial communications protocol called MIPI.</p> <p>The main reason is they take too many pins (hence space and cost) and are harder to work with at higher speeds.</p> |
| <p>Bus (computer bus)</p> | <p>A bus is a communication system that transfers data between components inside a computer, chips on a circuit board, or peripherals inside a chip.</p> <p>It's kind of synonymous with port, as in "serial port/serial bus", "parallel port/parallel bus", "I2C port/I2C bus", etc.</p> <p>It simply means a method of transporting data, like a literal bus that ferries data from one point to another.</p>  |
| <p>Bit</p> | <p>A bit, short for binary digit, is defined as the most basic unit of data in telecommunications and computing.</p> <p>Each bit is represented by either a 1 or a 0 and is physically expressed as voltage levels.</p> <p>Hence a 1 could be 3.3V (HIGH) and a 0 could be 0V (LOW).</p> <p>In the most basic sense, this is how information is expressed and transmitted in computing.</p> |

| | |
|---|--|
| Byte | <p>A byte is a storage unit capable of representing a single character, such as a letter, number or symbol.</p> <p>In most computers, one byte is equated to eight smaller units called bits and can represent 256 or 2^8 values.</p> <p>You'll find that in embedded computing where we work very near the level of the processor, we will be using a binary number system (base 2) often more than a decimal number system (base 10).</p> |
| Serial Peripheral Interface (SPI) | <p>SPI is an interface bus commonly used to send data between microcontrollers and small peripherals such as sensors, and SD cards.</p> |
| USB Serial Bridge (USB Dongle) | <p>Connects the USB port on the PC to the Serial port on the WildLogger.</p> <p>Allows the PC to communicate with the WildLogger.</p> |
| CH340 | <p>The chip (or integrated circuit) on the USB dongle. It's a popular USB to serial bridge that allows the PC to communicate with serial devices.</p> |
| Serial Console or Terminal / Serial Monitor | <p>A separate pop-up window in the Arduino IDE that allows you to both send messages from your computer to an Arduino board (over USB) and also to receive messages from the Arduino.</p> |
| GUI (Graphical User Interface) | <p>A graphical user interface (GUI) is an interface through which a user interacts with electronic devices such as computers, hand-held devices and other appliances.</p> <p>This interface uses icons, menus and other visual indicator (graphics) representations to display information and related user controls, unlike text-based interfaces, where data and commands are in text.</p> |
| Baudrate | <p>The speed at which data is transferred.</p> |

| | |
|-----------------------------|---|
| Bits per second (bps) | <p>Short for <i>bits</i> per second.</p> <p>bps is a measurement of the data transmission speed indicating of how many bits pass a certain point in one second.</p> <p>The higher the bps rate, the faster the download or upload time will be.</p> <p>Denoted with lower case b.</p> |
| Bytes per second (B/s, Bps) | <p>Short for <i>bytes</i> per second and also sometimes abbreviated as B/s,</p> <p>Bps is a speed that is measured in bytes transmitted each second.</p> <p>Denoted with uppercase B.</p> |
| Serial Bit Errors | <p>When a bit (data) gets lost or is corrupted.</p> <p>This can happen in serial communications when the transmitter and receiver aren't synchronized well.</p> |

Useful Links

| Video | URL |
|-------------------------------------|---|
| Arduino Desktop IDE | https://arduino.cc/download |
| Arduino Reference Language | https://www.arduino.cc/reference/en/ |
| WildLogger Board Description Files: | https://raw.githubusercontent.com/freaklabs/freaklabs-wildlife-boards/master/package_freaklabs_index.json |
| CH340 Driver | https://bit.ly/wildlogger-driver |
| Installing CH340 Drivers | https://learn.sparkfun.com/tutorials/how-to-install-ch340-drivers/all |

| | |
|----------------|---|
| WildLogger Kit | https://cutt.ly/freaklabs-wildlogger-kit |
|----------------|---|

Tutorials

| | |
|------------------|---|
| Olympia Circuits | Good overview of C++ used in Arduino http://learn.olympiacircuits.com/programming.html |
| Tutorials Point | More detailed tutorial on Arduino https://www.tutorialspoint.com/arduino/index.html |