IoT Devices

IoT

Raaspberry Pi

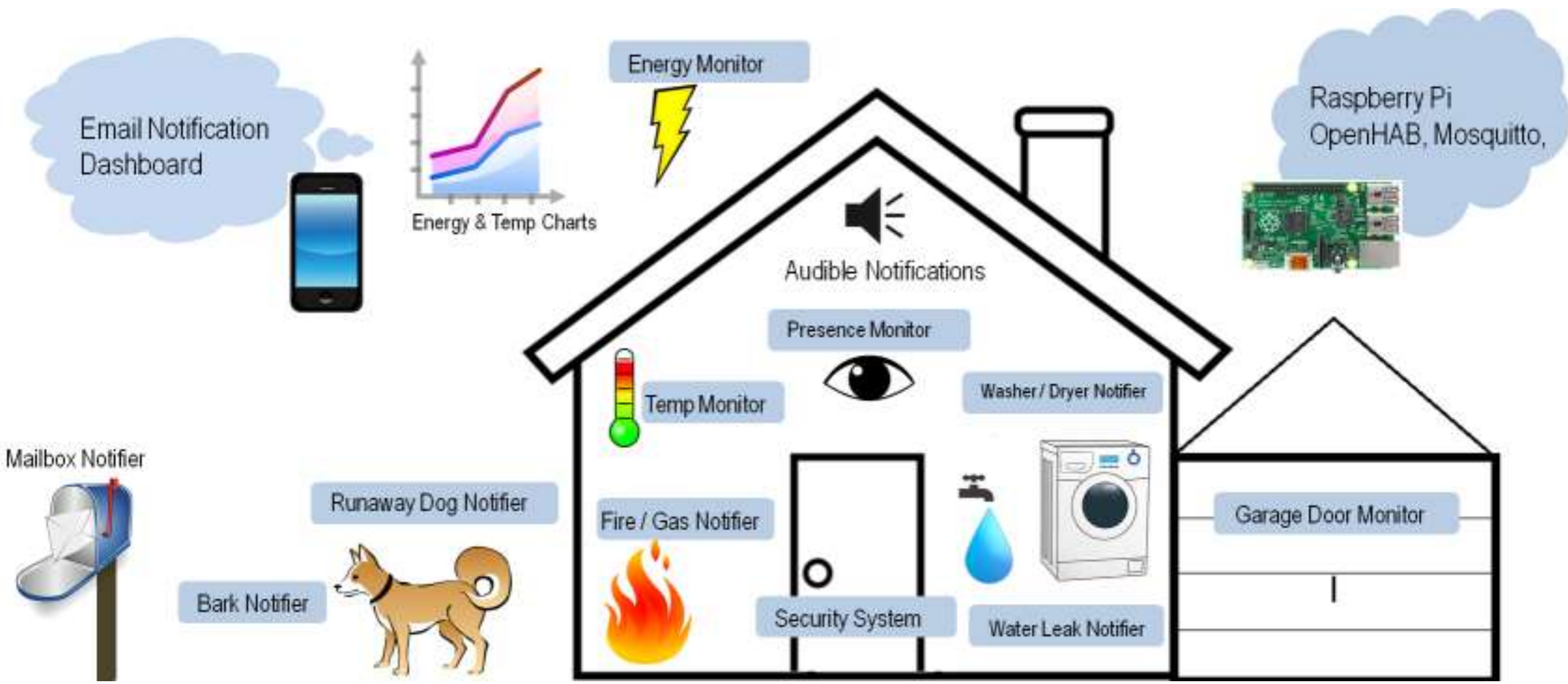
Arduino

BBC Micro:bit

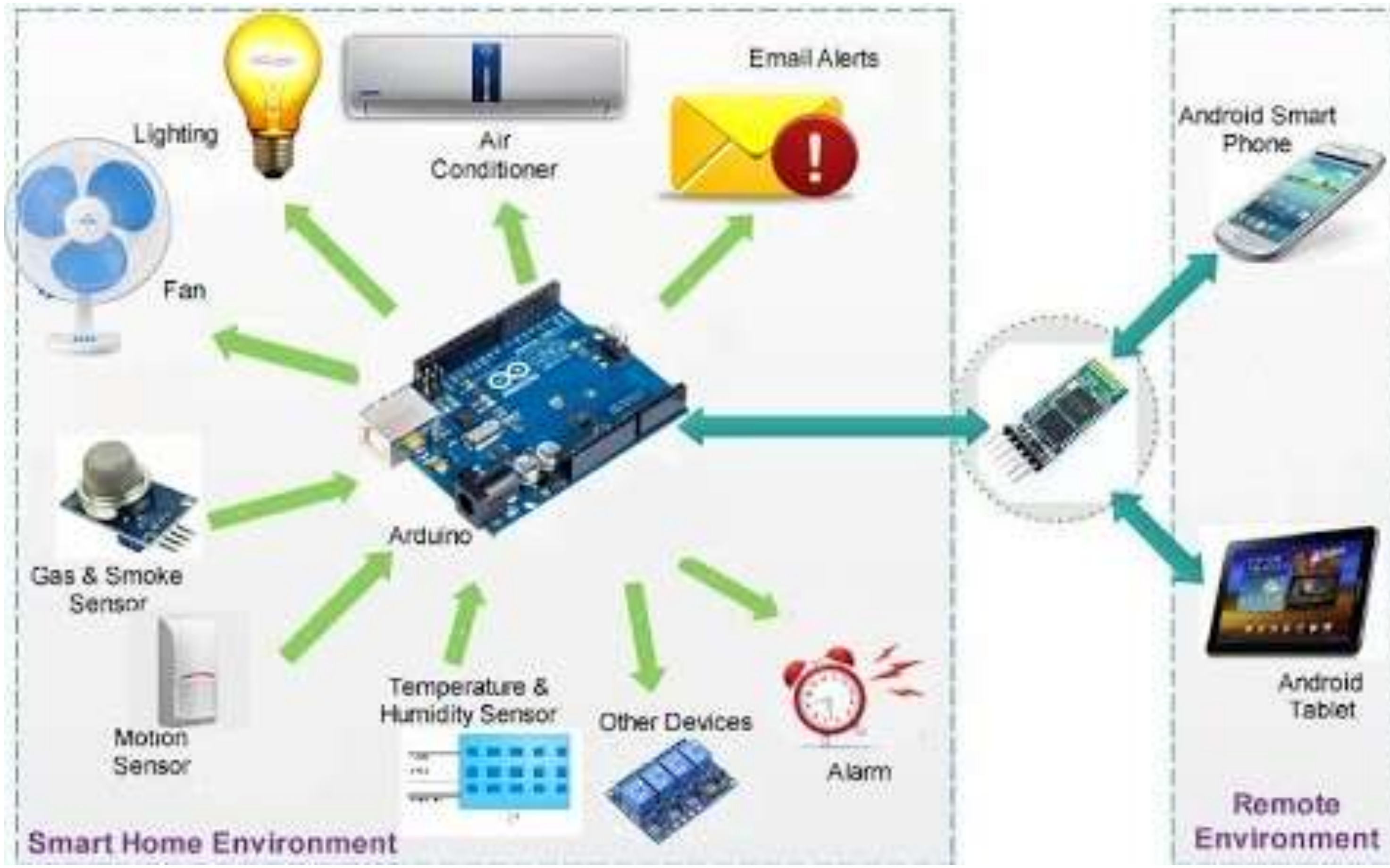
IoT – The Internet of Things

- The Internet of Things (IoT) is made possible because small computers can now be made very cheaply

- Many every day devices and appliances can thus include a small computer that can control the device and communicate over the internet.

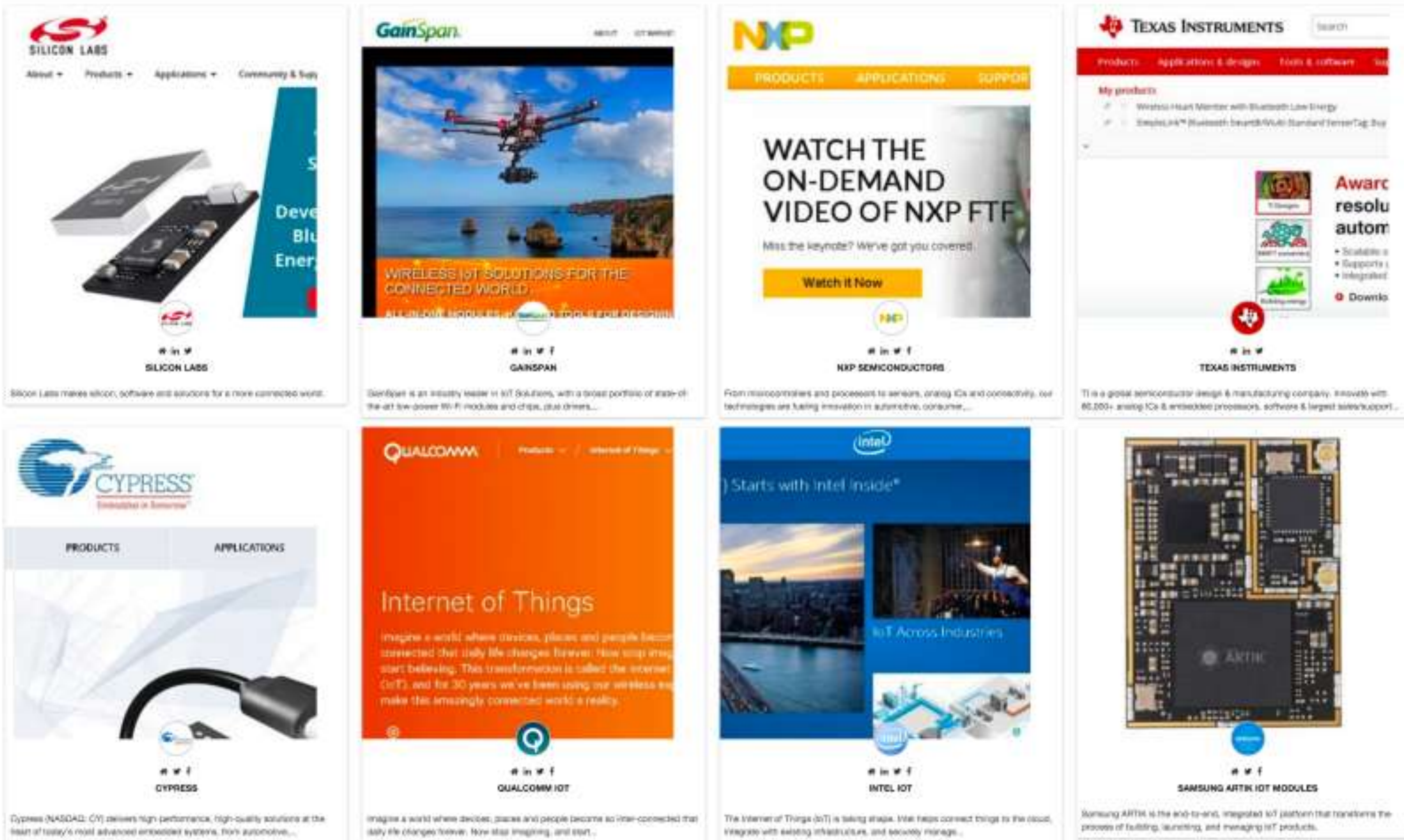


IoT at home



Lighting
Air Conditioner
Email Alerts
Fan
Android Smart Phone
Arduino
Gas & Smoke Sensor
Motion Sensor
Temperature & Humidity Sensor
Other Devices
Alarm
Android Tablet
Smart Home Environment
Remote Environment

Some IoT hardware providers



Common IoT devices

- IoT chipsets typically provide internet connectivity via WIFI (802xx) and or bluetooth, as well as a processor and memory, and some I/O ports.
  - There also exist systems on a small board to allow for prototyping and enthusiasts to build their own IoT systems
  - A large number of small single board computers now exist. The two most common are ***Rasperry Pi*** and ***Arduino***

Rasperry Pi

Arduino

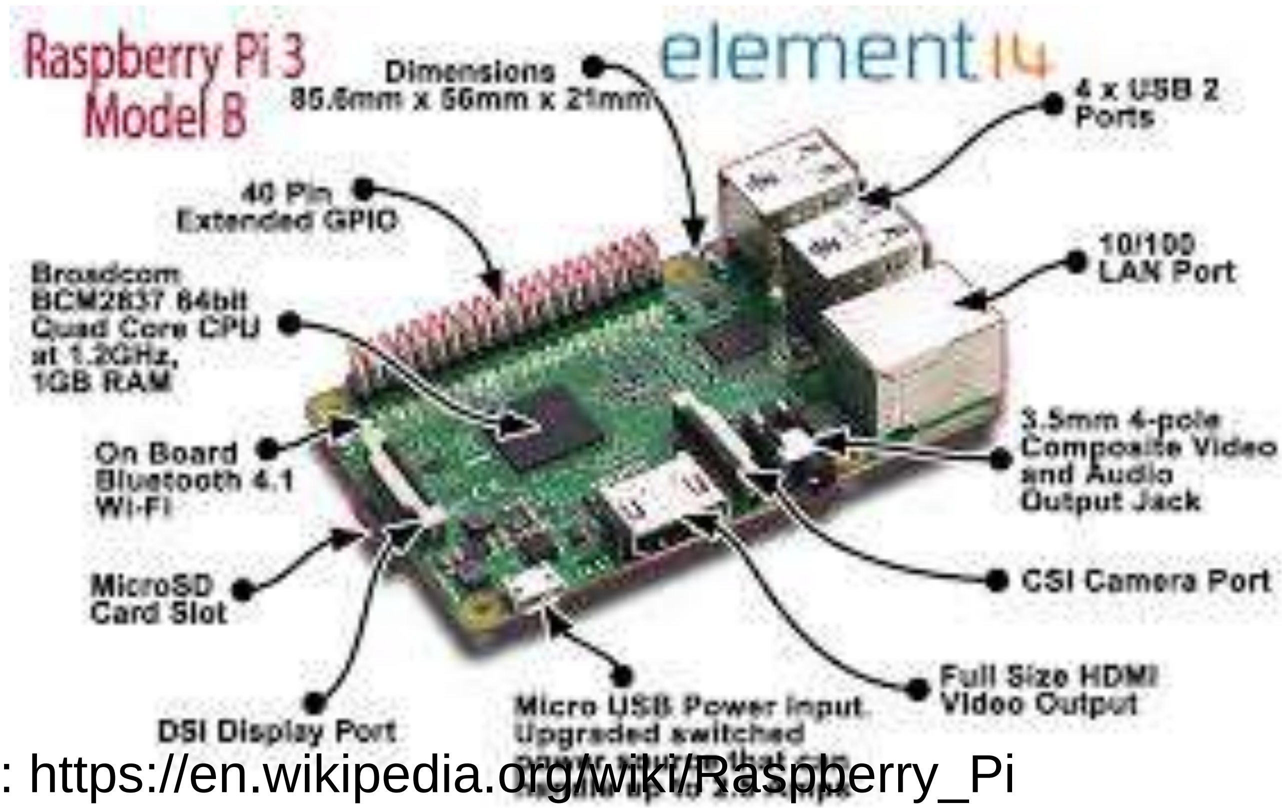
Raspberry Pi

- Model 1 B released in Feb 2012
- Model 1 A released shortly after - simpler
- Model B+ 2014

- Pi Zero released November 2015 smaller size for embedded applications

Pi 3 Model B, released Feb 2016.

Includes Wifi, Bluetooth. 1.2GHz quad core ARM Cortex-A53 CPU, GPU, 1GB SDRAM, HDMI, microSD, 4 USB ports, ~$70 on ebay.



Full details: https://en.wikipedia.org/wiki/Raspberry_Pi

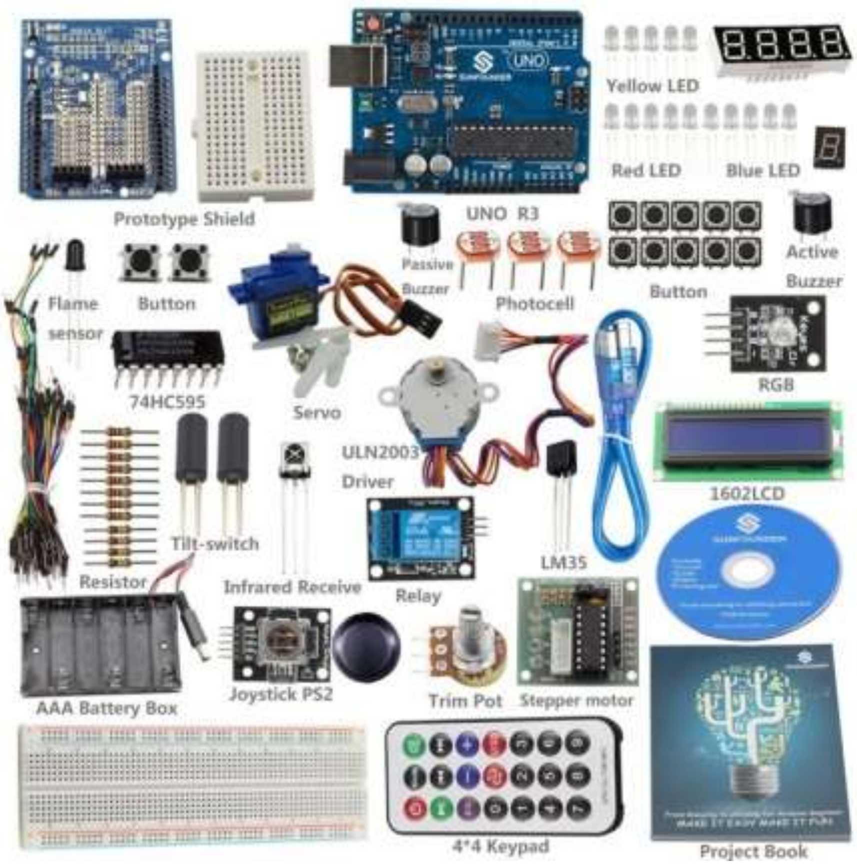
Arduino

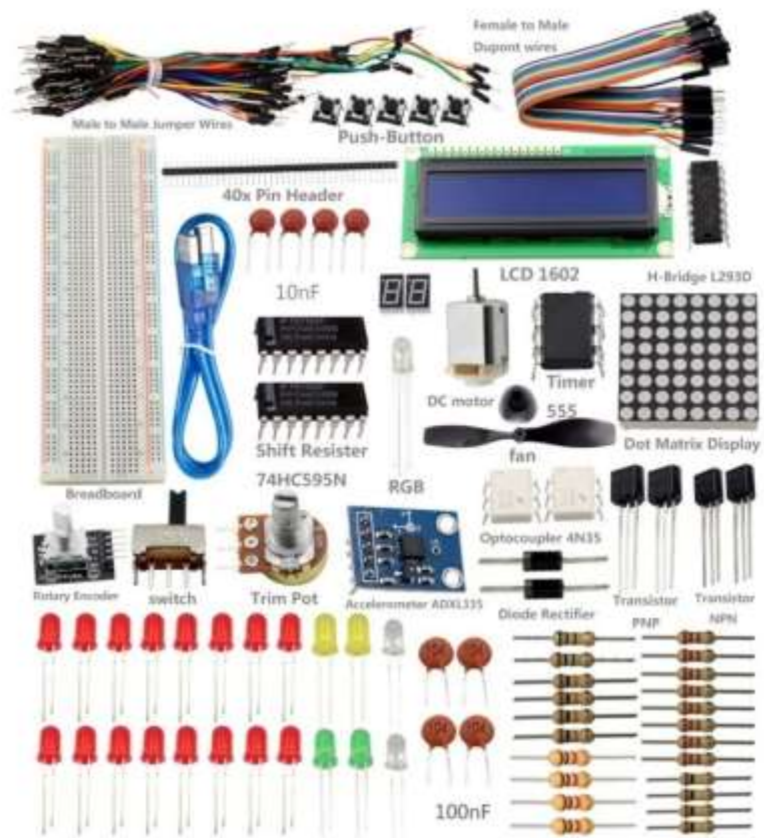
Arduino

- Developed in Italy to use for teaching about micro controllers.
  - Available commercially since around 2007
  - Mainly based on the Atmega(8/1632) series of processors. Typically 16Mhz
  - Typically 32k SDRAM, but varies from 2 to 256k
  - Large number of variations, in size of unit and functionality
  - Current “standard” board is Uno R3 about $5-$15 in Australia for board alone
  - Arduino is cheaper and simpler than Raspberry Pi
  - Extra capability can be achieved by adding electrical components, sensors, or “Shields”

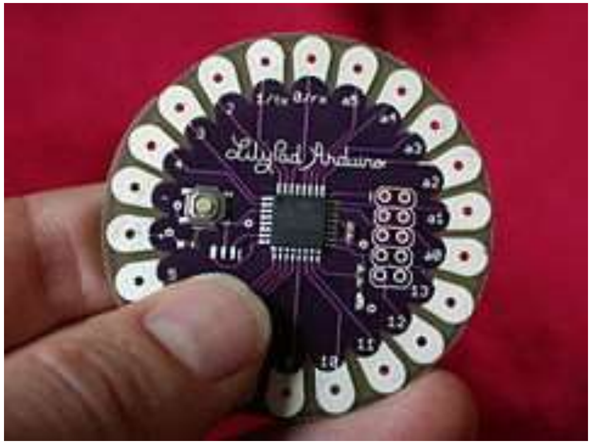
Loose sensors as part of a kit

Both Raspberry Pi and Arduino do not come with their own sensrs – you much buys them separately.

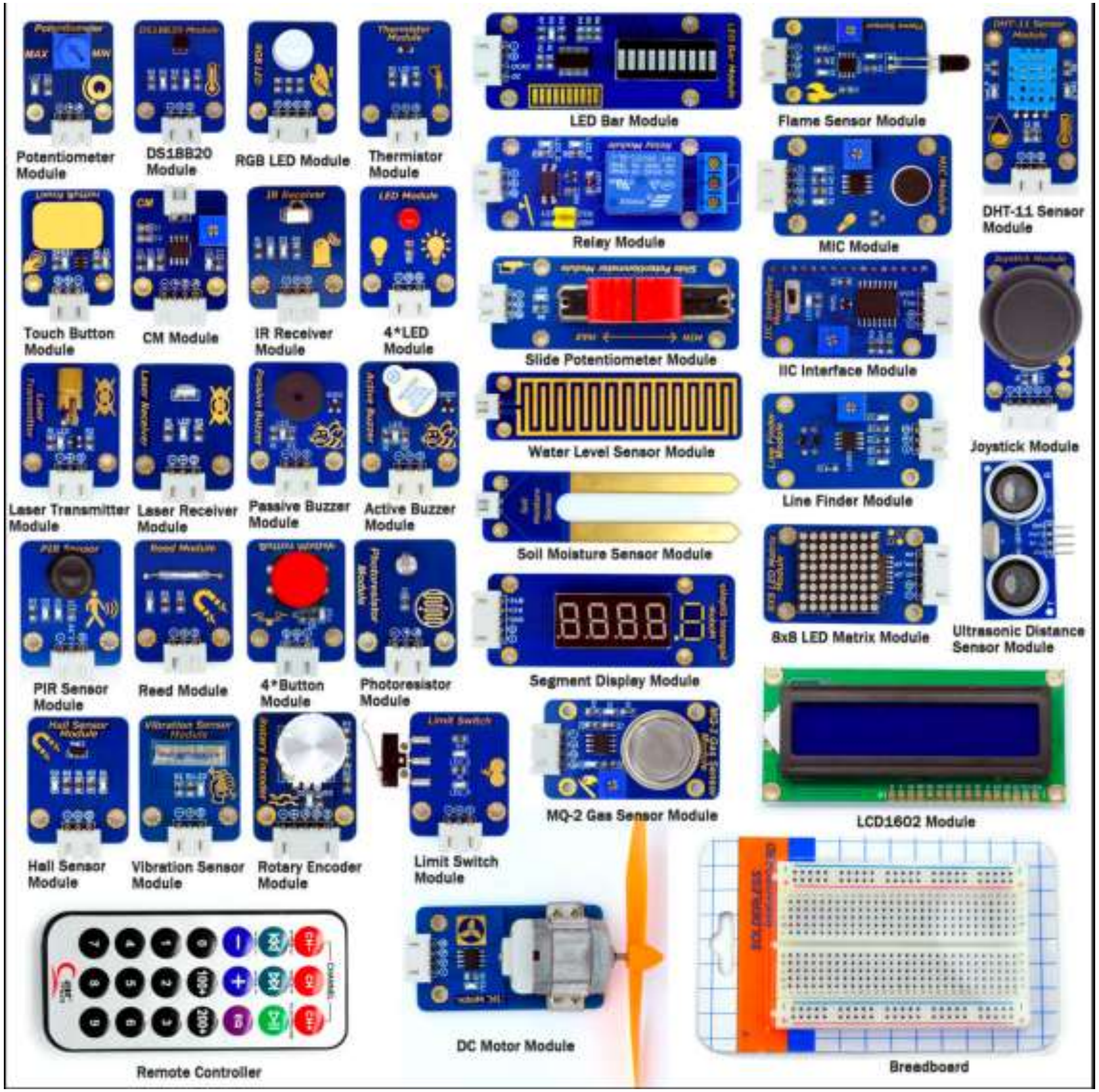




Arduino boards



Sensors on small boards



Sheilds

Arduino Main Features

- 13 Digital pins, These can be configured as input or output
  - You can use any of the pins to control devices or sensors, or to read digital input. Simply change the pin number constant in your code
- 5 Analog pins
  - These are used to read input from analog sensors. The Arduino has an inbuilt A/D converter that will convert the analog voltage to a number in the range 0-1023
- 5V, 3.3V and ground pins
  - To supply power as needed

Micro:bit – ARM architechure

- ARM stands for Advanced Risc Machine
- The ARM company has designed a range of processor cores and licenses them for use.

- ARM holdings was originally Acorn Computers and made a type of desktop computer in the 80s

- Arm Holdings develops the architecture and licenses it to other companies, who design their own products that implement one of those architectures—including systems-on-chips (SoC) and systems-on-modules (SoM) that incorporate memory, interfaces, radios, etc. (*Wikipedia*)

ARM Architecture

Architecture	Core bit-width	Cores	Profile	Refe-rences	
ARM Holdings	Third-party				
ARMv1	32[a 1]	ARM1		Classic	
ARMv2	32[a 1]	ARM2, ARM250, ARM3	Amber, STORM Open Soft Core[39]	Classic	
ARMv3	32[a 2]	ARM6, ARM7		Classic	
ARMv4	32[a 2]	ARM8	StrongARM, FA526, ZAP Open Source Processor Core[40]	Classic	
ARMv4T	32[a 2]	ARM7TDMI, ARM9TDMI, SecurCore SC100		Classic	
ARMv5TE	32	ARM7EJ, ARM9E, ARM10E	XScale, FA626TE, Feroceon, PJ1/Mohawk	Classic	
ARMv6	32	ARM11		Classic	
ARMv6-M	32	ARM Cortex-M0, ARM Cortex-M0+, ARM Cortex-M1, SecurCore SC000		Microcontroller	
ARMv7-M	32	ARM Cortex-M3, SecurCore SC300		Microcontroller	
ARMv7E-M	32	ARM Cortex-M4, ARM Cortex-M7		Microcontroller	
ARMv8-M	32	ARM Cortex-M23,[41] ARM Cortex-M33[42]		Microcontroller	[43]
ARMv7-R	32	ARM Cortex-R4, ARM Cortex-R5, ARM Cortex-R7, ARM Cortex-R8		Real-time	
ARMv8-R	32	ARM Cortex-R52		Real-time	[44][45][46]
ARMv7-A	32	ARM Cortex-A5, ARM Cortex-A7, ARM Cortex-A8, ARM Cortex-A9, ARM Cortex-A12, ARM Cortex-A15, ARM Cortex-A17	Qualcomm Krait, Scorpion, PJ4/Sheeva, Apple Swift	Application	
ARMv8-A	32	ARM Cortex-A32		Application	
ARMv8-A	64/32	ARM Cortex-A35,[47] ARM Cortex-A53, ARM Cortex-A57,[48] ARM Cortex-A72,[49] ARM Cortex-A73[50]	X-Gene, Nvidia Project Denver, Cavium Thunder X,[51][52][53] AMD K12, Apple Cyclone/Typhoon/Twister/Hurricane/Zephyr, Qualcomm Kryo, Samsung M1 and M2 ("Mongoose")[54]	Application	[55][56]
ARMv8.1-A	64/32	TBA	ThunderX2[57]	Application	
ARMv8.2-A	64/32	ARM Cortex-A55,[58] ARM Cortex-A75,[59] ARM Cortex-A76[60]		Application	[61]
ARMv8.3-A	64/32	TBA	Apple A12 Bionic	Application	
ARMv8.4-A	64/32	TBA		Application	

ARM Processors

- High end ARM processors are used extensively in mobile phones and tablets

- The Cortex range is intended for use in micro-controllers
- ARM based processors are used in at least 60% of mobile devices

- There are many different versions: M0-M35
- The Micro:bit is based on the ARM Cortex-M0 mircro controller

ARM Cortex processors

ARM Cortex-M instruction variations									
Arm Core	Cortex M0[2]	Cortex M0+[3]	Cortex M1[4]	Cortex M3[5]	Cortex M4[6]	Cortex M7[7]	Cortex M23[8]	Cortex M33[12]	Cortex M35P
ARM architecture	ARMv6-M[9]	ARMv6-M[9]	ARMv6-M[9]	ARMv7-M[10]	ARMv7E-M[10]	ARMv7E-M[10]	ARMv8-M Baseline[15]	ARMv8-M Mainline[15]	ARMv8-M Mainline[15]
Computer architecture	Von Neuman	Von Neumann	Von Neumann	Harvard	Harvard	Harvard	Von Neumann	Harvard	Harvard
Instruction pipeline	3 stages	2 stages	3 stages	3 stages	3 stages	6 stages	2 stages	3 stages	3 stages
Thumb-1 instructions	Most	Most	Most	Entire	Entire	Entire	Most	Entire	Entire
Thumb-2 instructions	Some	Some	Some	Entire	Entire	Entire	Some	Entire	Entire
Multiply instructions 32x32 = 32-bit result	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Multiply instructions 32x32 = 64-bit result	No	No	No	Yes	Yes	Yes	No	Yes	Yes
Divide instructions 32/32 = 32-bit quotient	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
Saturated instructions	No	No	No	Some	Yes	Yes	No	Yes	Yes
DSP instructions	No	No	No	No	Yes	Yes	No	Optional	Optional
Single-Precision (SP) Floating-point instructions	No	No	No	No	Optional	Optional	No	Optional	Optional
Double-Precision (DP) Floating-point instructions	No	No	No	No	No	Optional	No	No	No
TrustZone instructions	No	No	No	No	No	No	Optional	Optional	Optional
Co-processor instructions	No	No	No	No	No	No	No	Optional	Optional
Interrupt latency (if zero-wait state RAM)	16 cycles	15 cycles	23 for NMI 26 for IRQ	12 cycles	12 cycles	12 cycles	15 no security ext 27 security ext	TBD	TBD

The Cortex-M0 core is optimized for small silicon die size and use in the lowest price chips.

- Key features of the Cortex-M0 core are:
- ARMv6-M architecture
- 3-stage pipeline
- 16 32 bit registers (PC is reg 15)
- Instruction sets:
  - Thumb-1 (most), missing CBZ, CBNZ, IT
  - Thumb-2 (some), only BL, DMB, DSB, ISB, MRS, MSR
  - 32-bit hardware integer multiply with 32-bit result

1 to 32 interrupts, plus NMI

- Memory: the micro:bit has 256k of flash memory and 16Kb of static RAM
- Other features: https://microbit.org/guide/features/

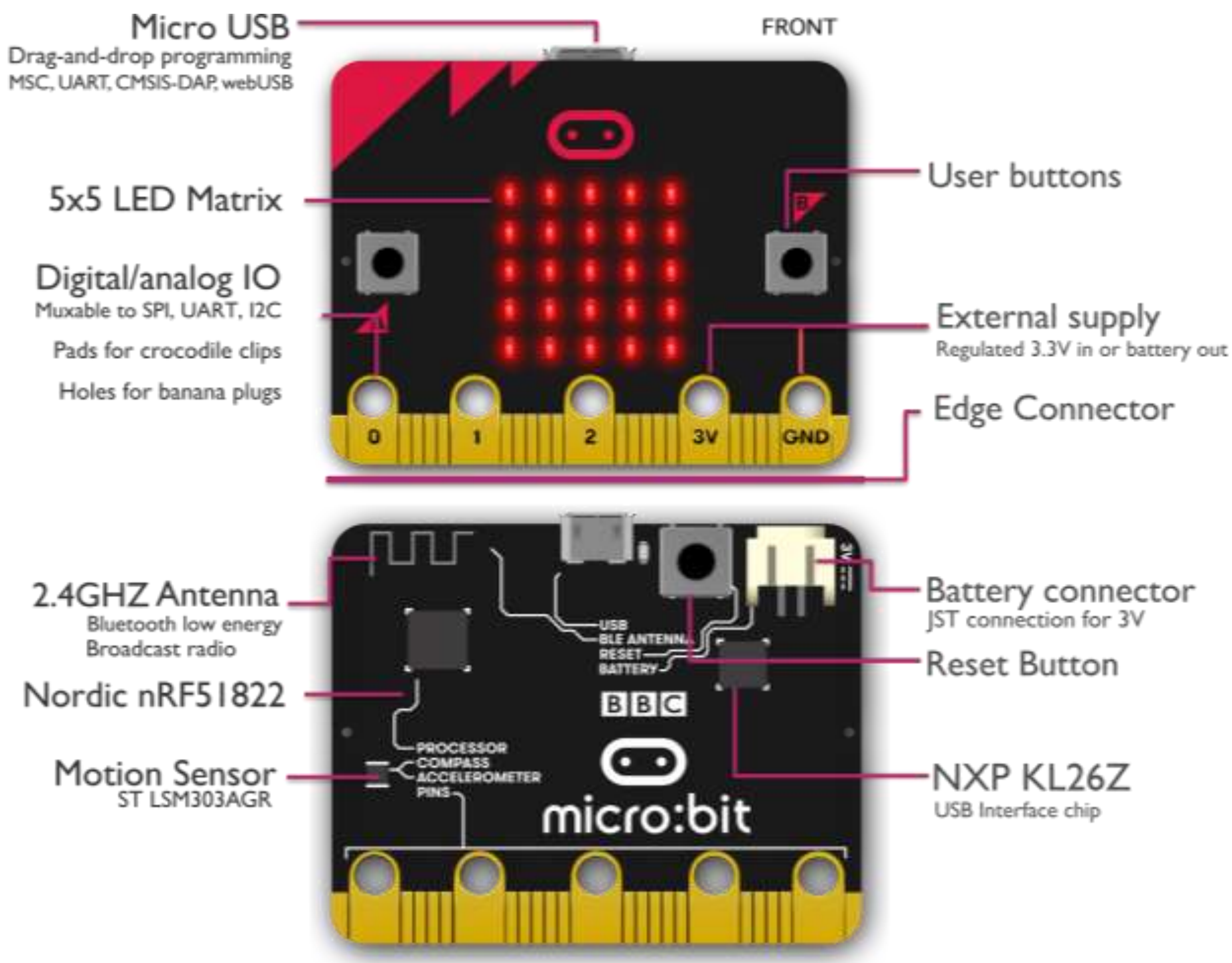
BBC Microbit

- **Much lower power than the Raspberry Pi or Arduino, , but look at all the sensors**
- **Biggest advantage:**
  - **Access to sensors, both standalone, and via PC**

Sensor suchs as :

- **Radio & Bluetooth antenna**
- **Processor & temperature sensor**
- **Compass**
- **Accelerometer and Gyroscope**
- **Input/Output pins**
- **Single and 25x LED for display and light sensing**
- **Reset button + 2 user Buttons**
- **Micro USB socket**
- **Battery socket**
- **USB interface chip**

BBC Microbit



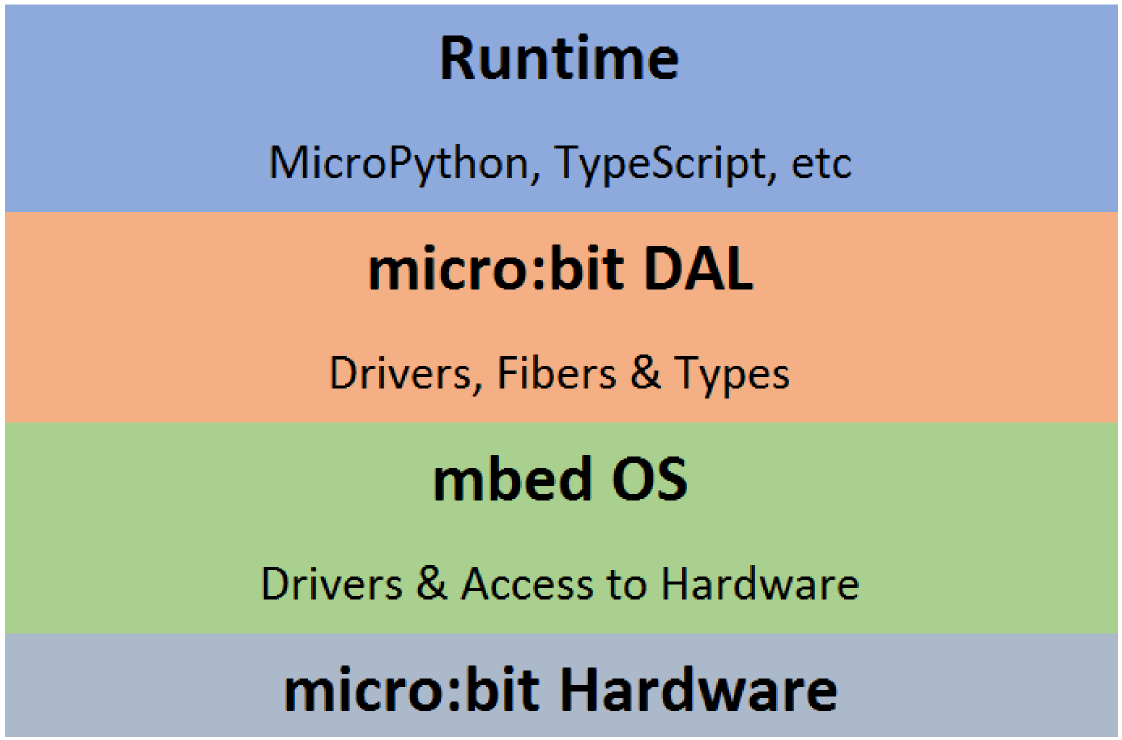
BBC Microbit – more detail

- When you connect the Micro:bit (MB) via USB, you will see an extra drive with some files on it.
  - If you hold down the reset button while connecting the USB, you will see a different set of files – including device.txt which contains the version numbers of all the software installed.

```
# DAPLink Firmware - see https://mbed.com/daplink
Unique ID: 0000000051864e45000a10070000004a0000000097969972
HIC ID: 97969901
Auto Reset: 0
Automation allowed: 1
Overflow detection: 0
Daplink Mode: Bootloader
Bootloader Version: 0243
Interface Version: 0249
Git SHA: b403a07e3696cee1e116d44cbdd64446e056ce38
Local Mods: 0
USB Interfaces: MSD
Bootloader CRC: 0x32eb3cfd
Interface CRC: 0xcdb7b2a3
Remount count: 0
```

Microbit Memory Map

- What is actually inside the ,e,pry pf the MB?



Microbit Memory Map

- **Runtime**
  - MicroPython, TypeScript and other languages

- **Microbit Device Abstraction Layer (DAL)**

Core – High-level components,

Types – Helper types such as ManagedString, Image, Event and PacketBuffer

Drivers – For control of a specific hardware component, such as Accelerometer, Button, Compass, Display, Flash, IO, Serial and Pin

Bluetooth – All the code for the Bluetooth Low Energy (BLE) stack that is shipped with the micro:bit

Microbit Memory Map

- **”mbed” Operating System (https://mbed.com)**
  - The software at the bottom of the stack is making use of the ARM mbed OS which is:
    - An open-source embedded operating system designed for the “things” in the Internet of Things (IoT).
    - mbed OS includes the features you need to develop a connected product using an ARM Cortex-M microcontroller.
    - mbed OS provides a platform that includes:
      - Security foundations.
      - Cloud management services.
      - Drivers for sensors, I/O devices and connectivity.
  - mbed OS is modular, configurable software that you can customize to your device and to reduce memory requirements by excluding unused software.

Microbit Memory Map

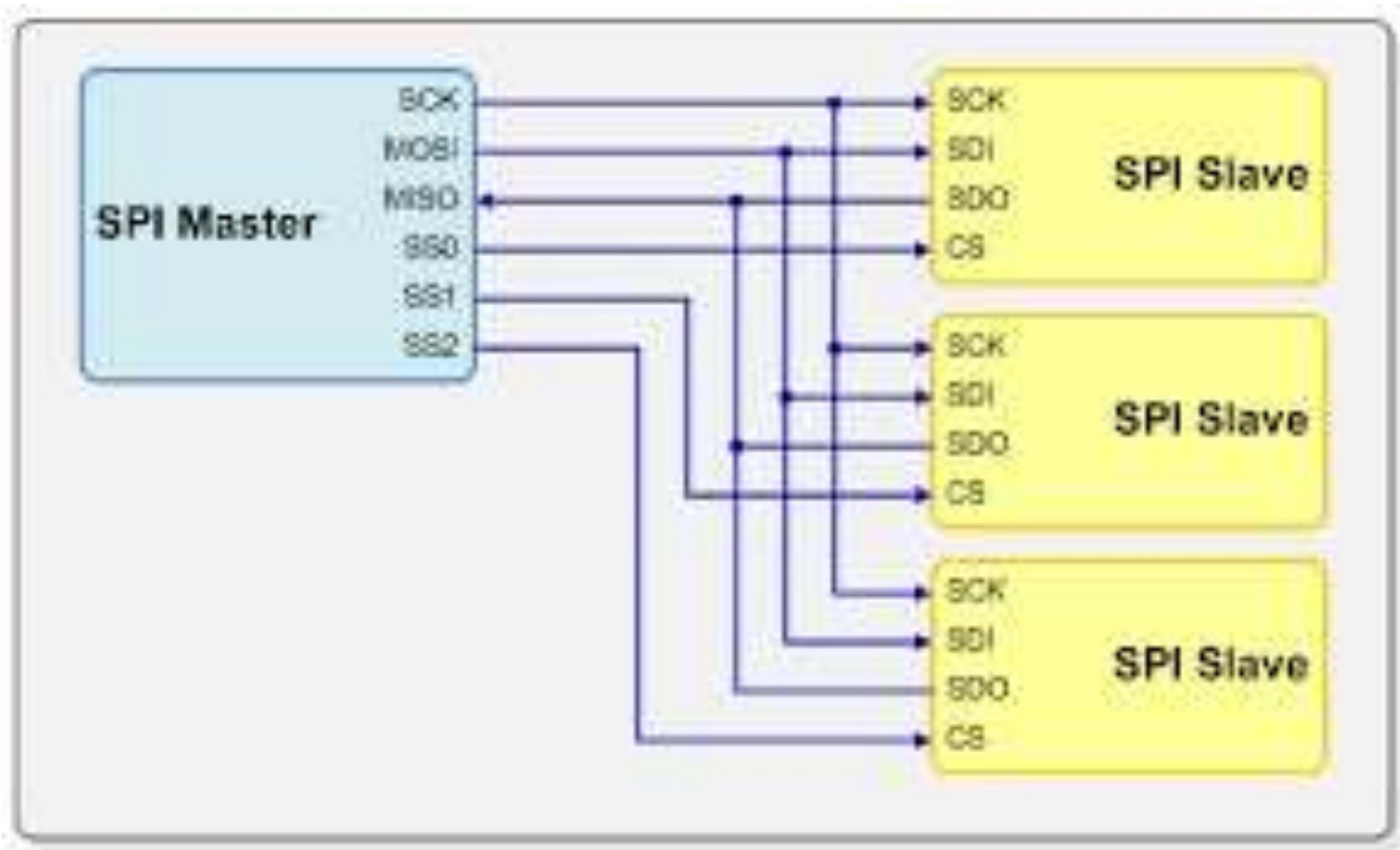
- **”mbed”. Operating System**
  - Lower level Device Drivers
  - Controlled Access
  - Security

- **Hardware**
  - **3D Accelerometer, Gyro**
  - **Compass**
  - **25 LED’s**
  - Pins

- More detail: https://mattwarren.org/2017/11/28/Exploring-the-BBC-microbit-Software-Stack/

Serial Peripheral Interface (SPI) Bus

- An industry standard bus specification that allows communication with small devices and sensors



SPI bus cont

- Each device (slave) has 4 connections
  - SCLK: Serial Clock (output from master).
  - MOSI: Master Output Slave Input, or Master Out Slave In (data output from master).
  - MISO: Master Input Slave Output, or Master In Slave Out (data output from slave).
  - SS: Slave Select (often active low, output from master).
- The master must have a separate select line for each device (called slave select on the Master, and Chip select on the slave)

- A programmer does not need to be too concerned with this. There are library functions to manage the bus, and libraries for each type of device