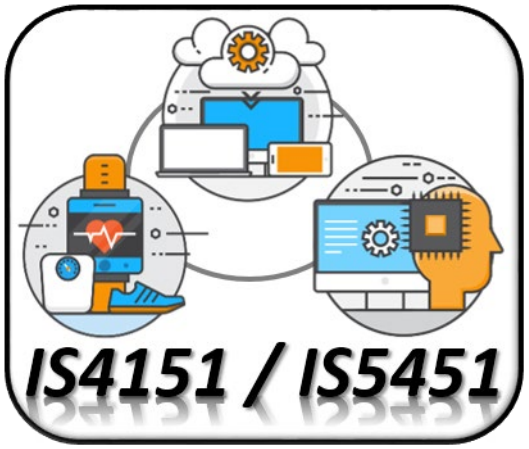


Lecture 5

Single-board Microcontroller (II)

IS4151/IS5451 – AIoT Solutions and Development
AY 2024/25 Semester 2

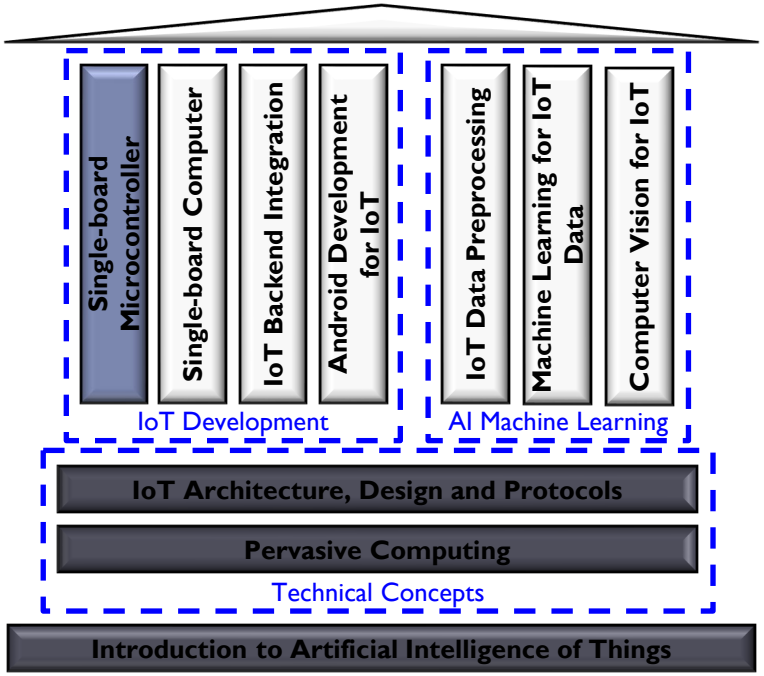
Lecturer: A/P TAN Wee Kek

Email: tanwk@comp.nus.edu.sg :: **Tel**: 6516 6731 :: **Office**: COM3-02-35

Consultation: Tuesday, 2 pm to 4 pm. Additional consultations by appointment are welcome.

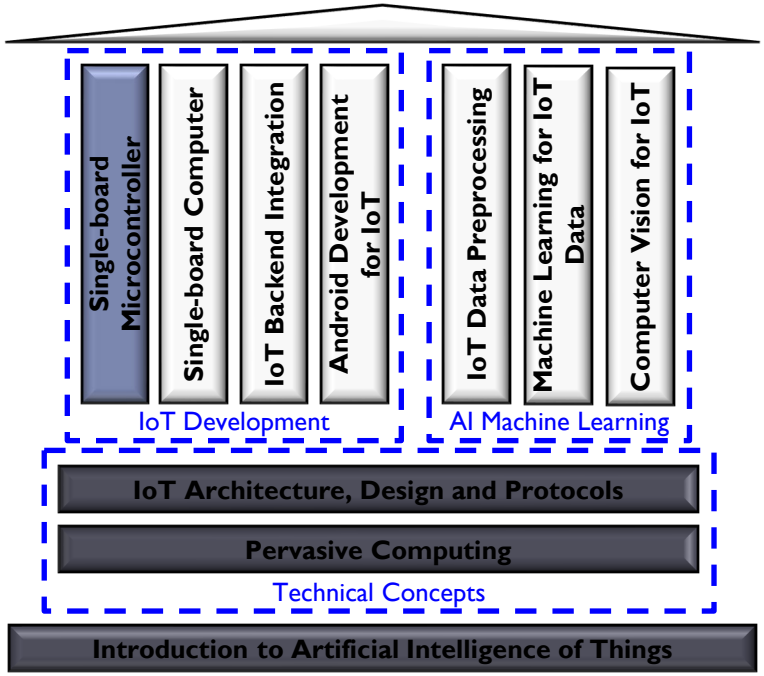
Quick Recap…

- In the previous lecture, we learnt:
  - That micro:bit is a single-board microcontroller with good computational and wireless data communication capabilities.
  - About the various sensors onboard micro:bit that enable it to sense and react to its environment.
- Micro:bit can be used to build a node device, but the onboard hardware capabilities are still limited:
  - What happens if the sensors that we need are not available?
  - How do we perform hard actuation?



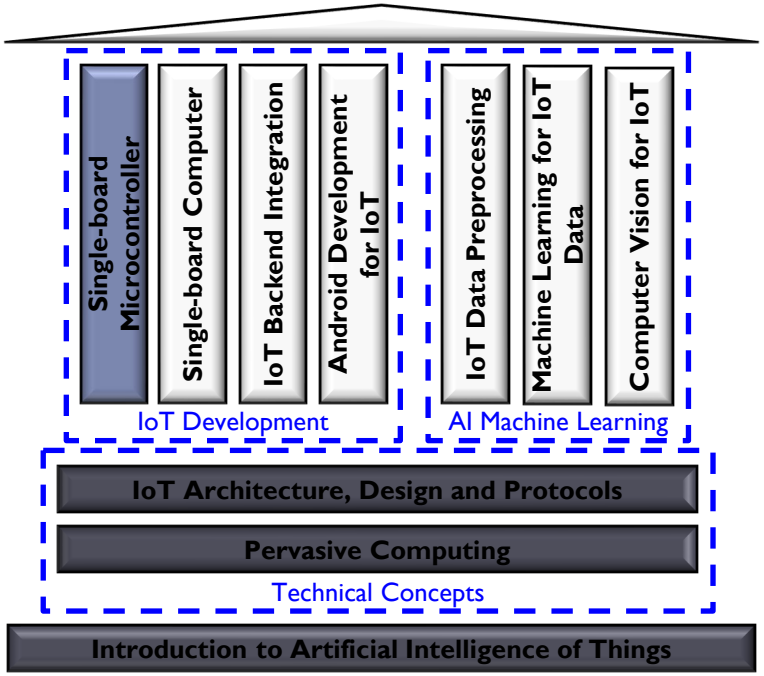
Quick Recap… (cont.)

- This lecture continues our learning journey to find out how to work with external sensors and actuators.



Learning Objectives

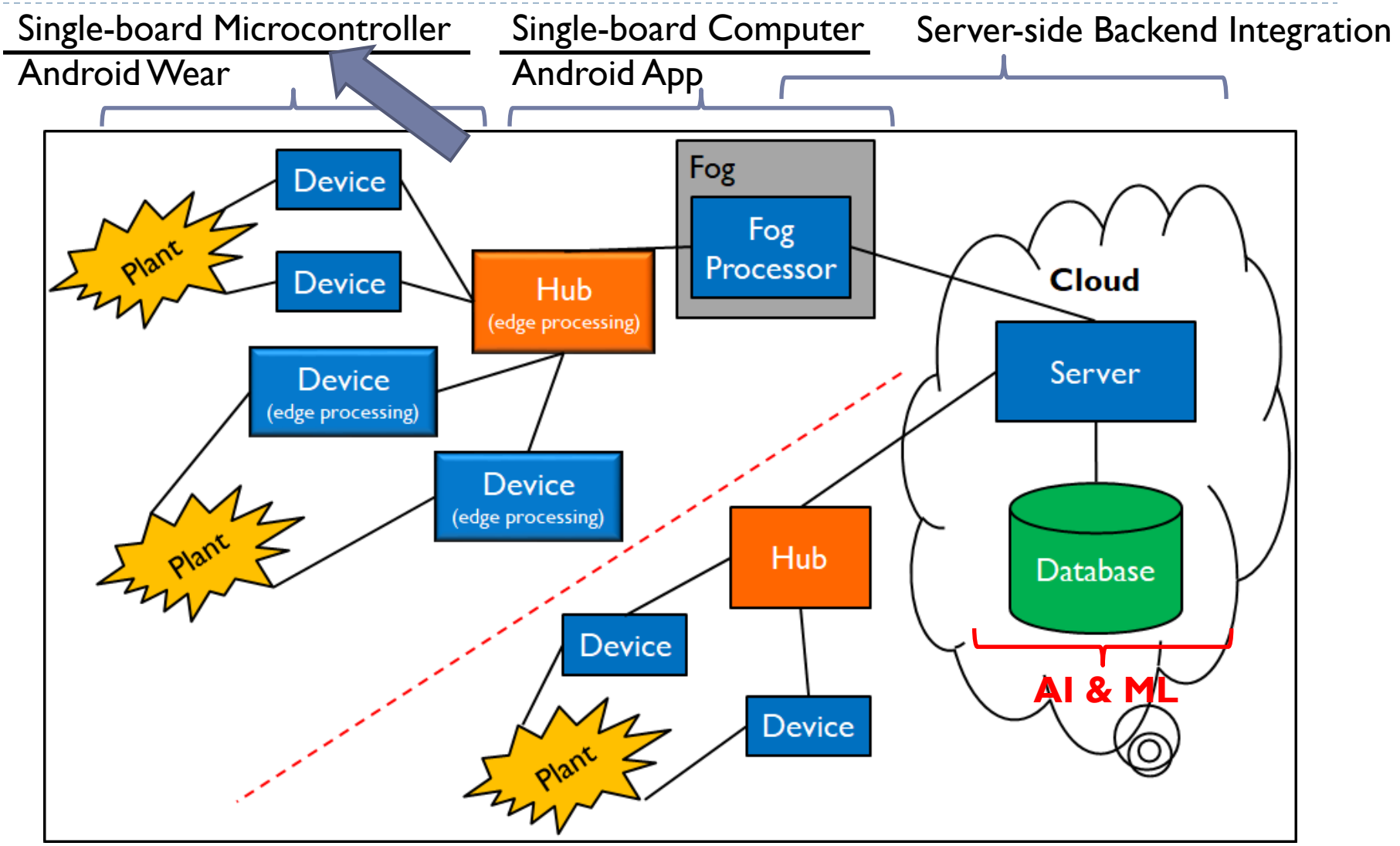
- At the end of this lecture, you should understand:
  - How to work with external peripherals via the micro:bit's edge connector.
  - What is an Analogue to Digital Converter.
  - How to write and read analogue and digital values to/from the micro:bit.
  - Programming peripherals with:
    - micro:bit's pins directly.
    - Edge connector breakout board.
    - Shield device.



Readings

- Required readings:
  - None.
- Suggested readings:
  - None.

Technical Roadmap for IS4151/IS5451

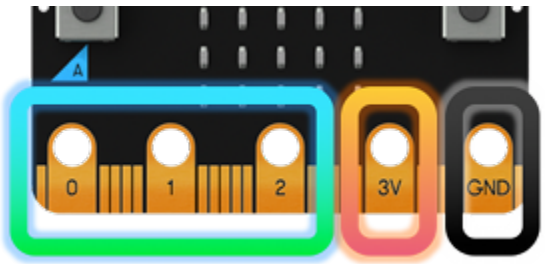


Micro:bit's Pins

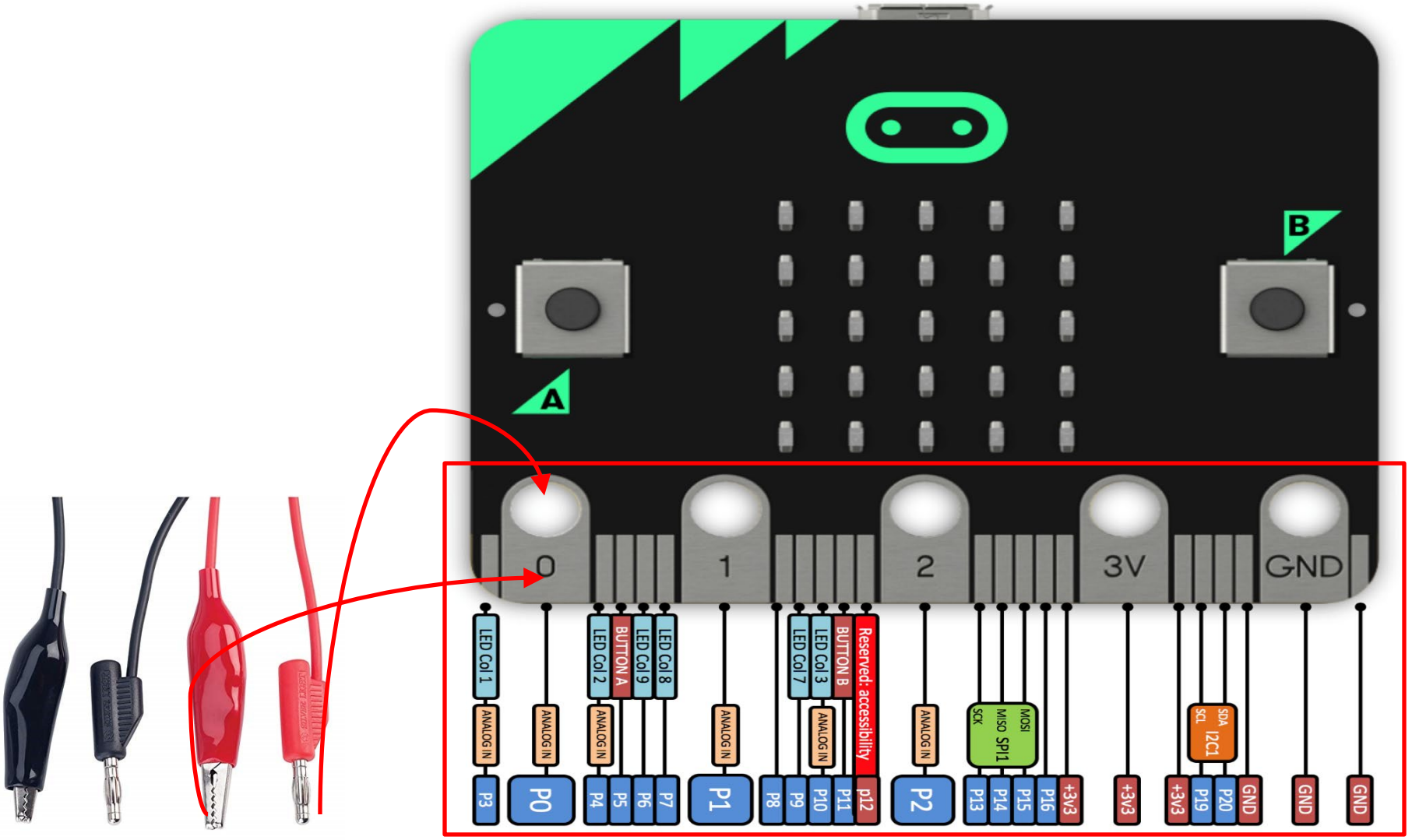
- Recall that there are 25 external connections on the edge connector of micro:bit:
  - These are known as the "pins".
  - They are used to program LEDs, motors or other electrical components → Output.
  - The "pins" can also be used to connect extra sensors to control our code → Input.
- There are five large pins, that are also connected to holes, on the board labelled – 0, 1, 2, 3V, and GND.
- Along the same edge, there are an additional 20 small pins that we can use when micro:bit is plugged into an edge connector breakout board.

More about Micro:bit's Large Pins

- Can easily attach crocodile clips or 4 mm banana plugs to the large pins.
- The first three large pins, labelled 0, 1 and 2 are flexible and can be used for many different purposes.
- They are often called "**G**eneral **P**urpose **I**nput and **O**utput" (shortened to **GPIO**) pins.
- These three pins also have the ability to read analogue voltages using an analogue-to-digital converter (ADC).
- Thus, the three large pins have the same function, i.e., GPIO with ADC.



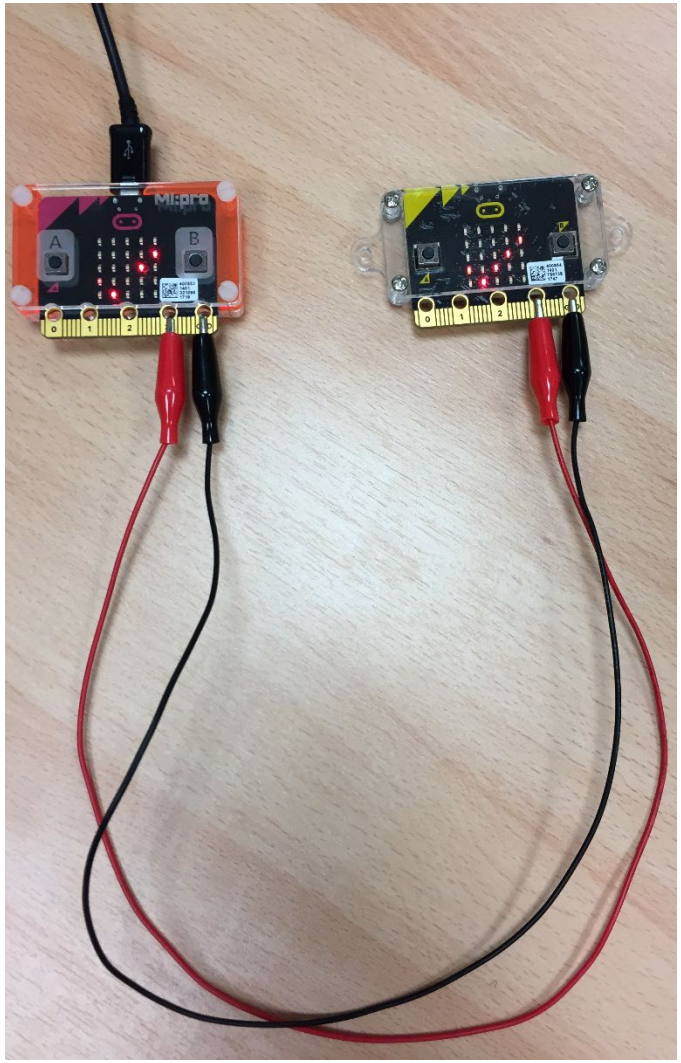
More about Micro:bit's Large Pins (cont.)

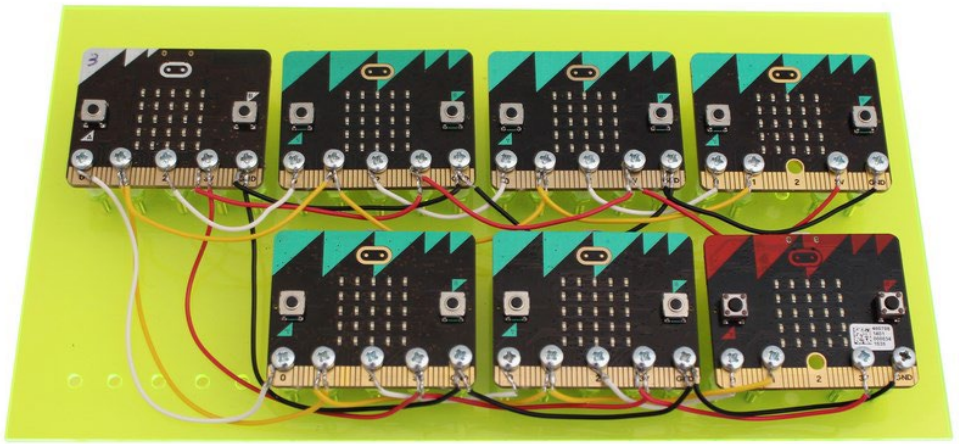


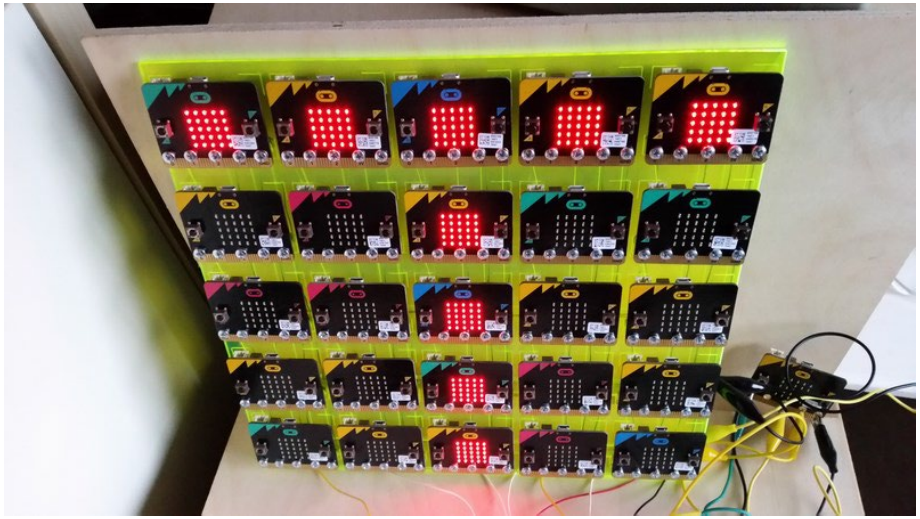
More about Micro:bit's Large Pins (cont.)

- The other two large pins (3V and GND) are very different.
- 3V:
  - 3 volt power output or power input.
  - **Power output** – If the micro:bit is powered by USB or a battery, then you can use the 3V pin as a power output to power other peripherals.
  - **Power input** – If the micro:bit is not being powered by USB or battery, you can use the 3V pin as a power input to power the device itself.
- GND:
  - Attaches to ground in order to complete a circuit (required when using the 3V pin).

More about Micro:bit's Large Pins (cont.)







How many micro:bit devices can be daisy chained together?

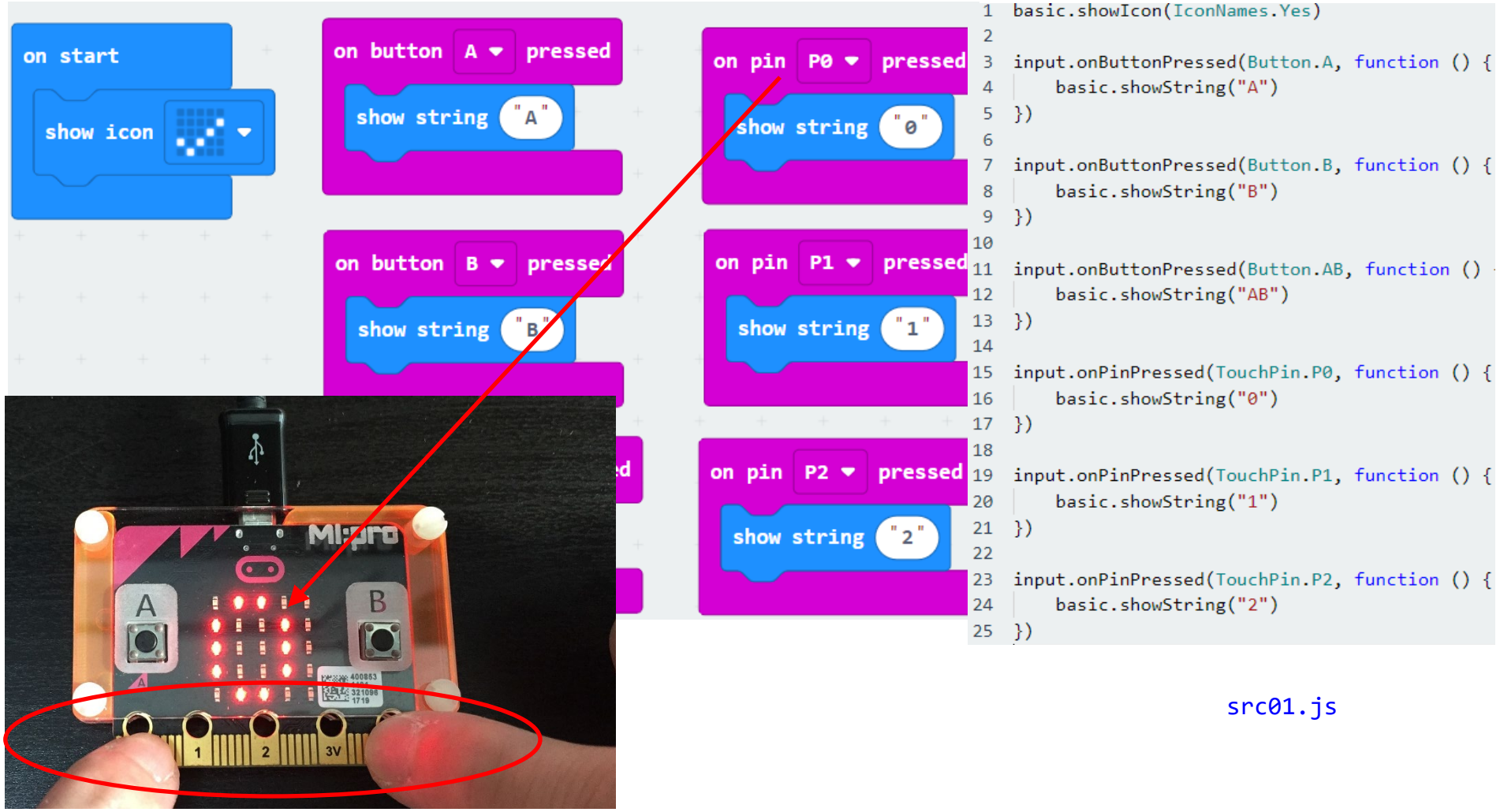
More about Micro:bit's Large Pins (cont.)

This display is made from 1,000 micro:bit devices by Kitronik!

More about Micro:bit's Large Pins (cont.)

- Extra buttons with the 3 large GPIO pins:
  - Hold the "GND" pin with one hand.
  - Program micro:bit to detect the touching of the 0,1 or 2 large GPIO pins with the other hand.
  - This gives you three more buttons to experiment with.
  - Basically, we use our body to complete an electrical circuit.

More about Micro:bit's Pins (cont.)

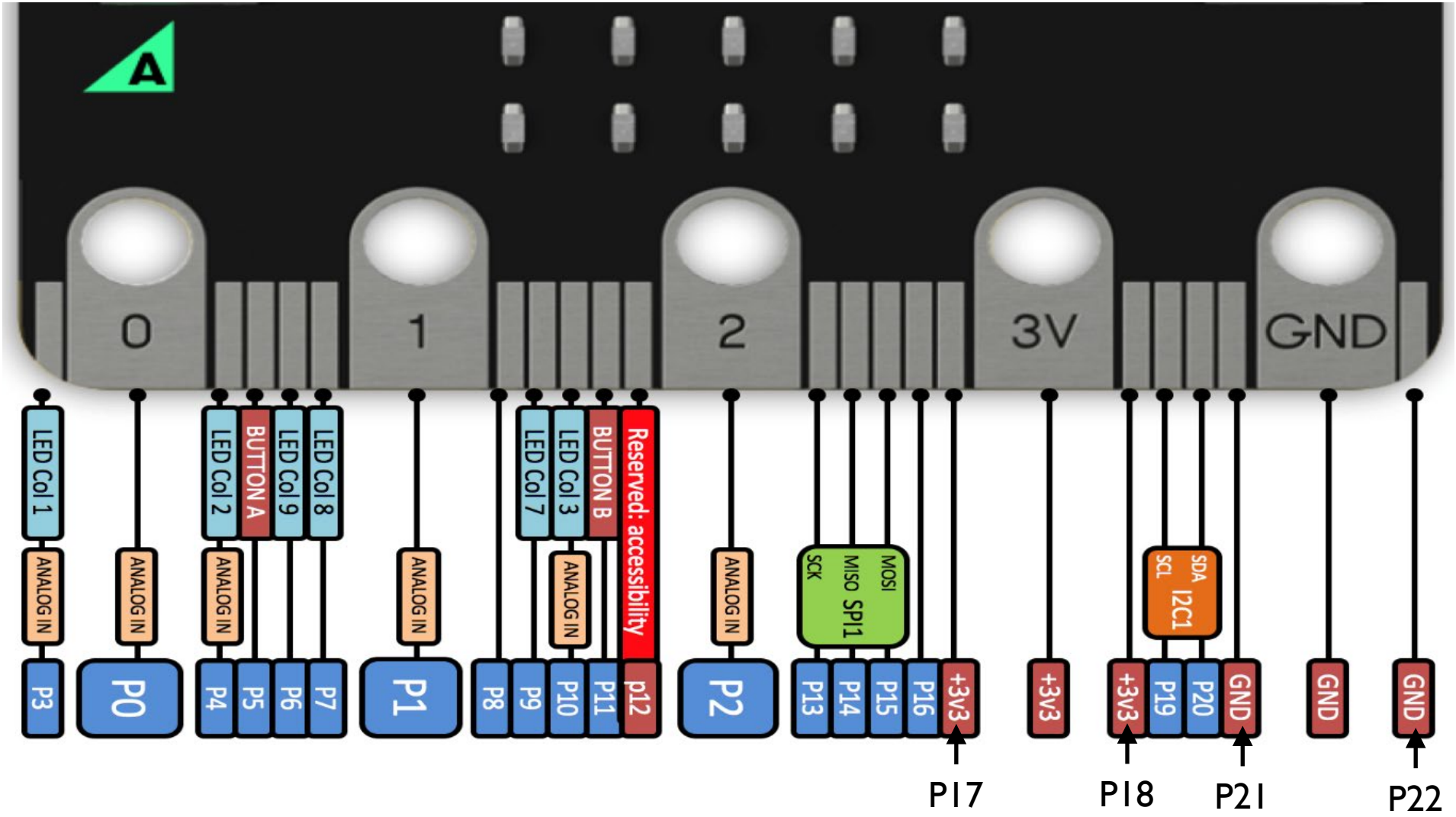


src01.js

More about Micro:bit's Small Pins

- The small pins are numbered sequentially from 3-22.
- They are not labelled on the micro:bit but they are labelled in the picture on the next slide for ease of reference.
- Unlike the three large pins that are dedicated to being used for external connections, some of the small pins are shared with other components on the micro:bit board:
  - E.g., Pin 3 is shared with some of the LEDs on the screen of the Micro:bit.
  - So, if you are using the screen to scroll messages, you cannot use this pin concurrently.

More about Micro:bit's Small Pins (cont.)

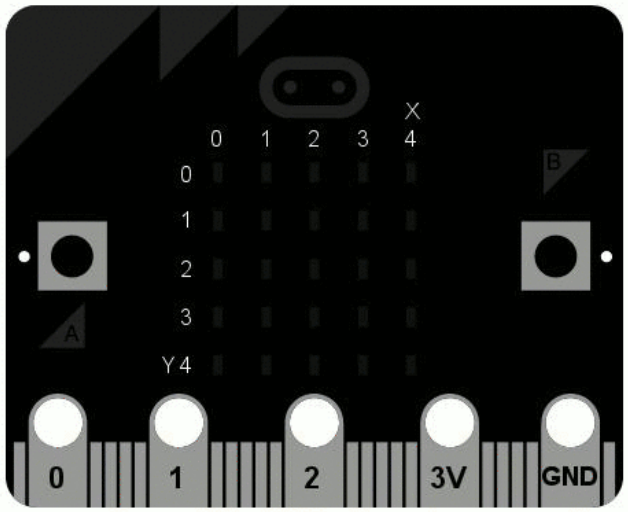


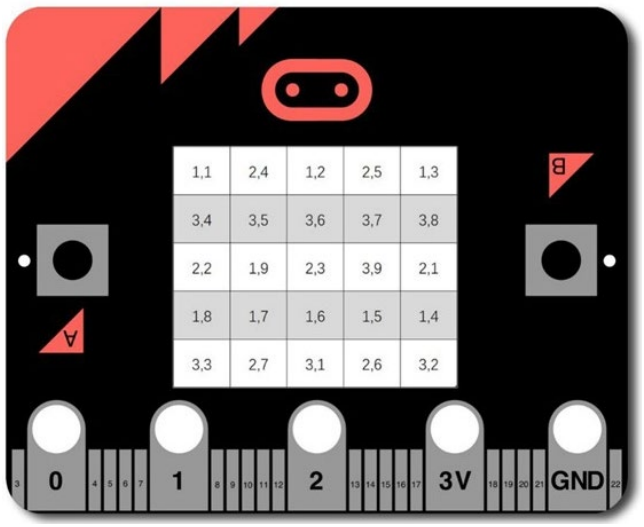
More about Micro:bit's Small Pins (cont.)

Pin	Use
Pin 3	GPIO shared with LED Col 1 of the LED screen; can be used for ADC and digital I/O when the LED screen is turned off.
Pin 4	GPIO shared with LED Col 2 of the LED screen; can be used for ADC and digital I/O when the LED screen is turned off.
Pin 5	GPIO shared with Button A. This lets you trigger or detect a button "A" click externally. This pin has a pull-up resistor, which means that by default it is at voltage of 3V. To replace button A on the BBC micro:bit with an external button, connect one end of the external button to pin 5 and the other end to GND. When the button is pressed, the voltage on pin 5 is pulled down to 0, which generates a button click event.
Pin 6	GPIO shared with LED Col 9 of the LED screen; can be used for digital I/O when the LED screen is turned off.
Pin 7	GPIO shared with LED Col 8 of the LED screen; can be used for digital I/O when the LED screen is turned off.
Pin 8	Dedicated GPIO, for sending and sensing digital signals.

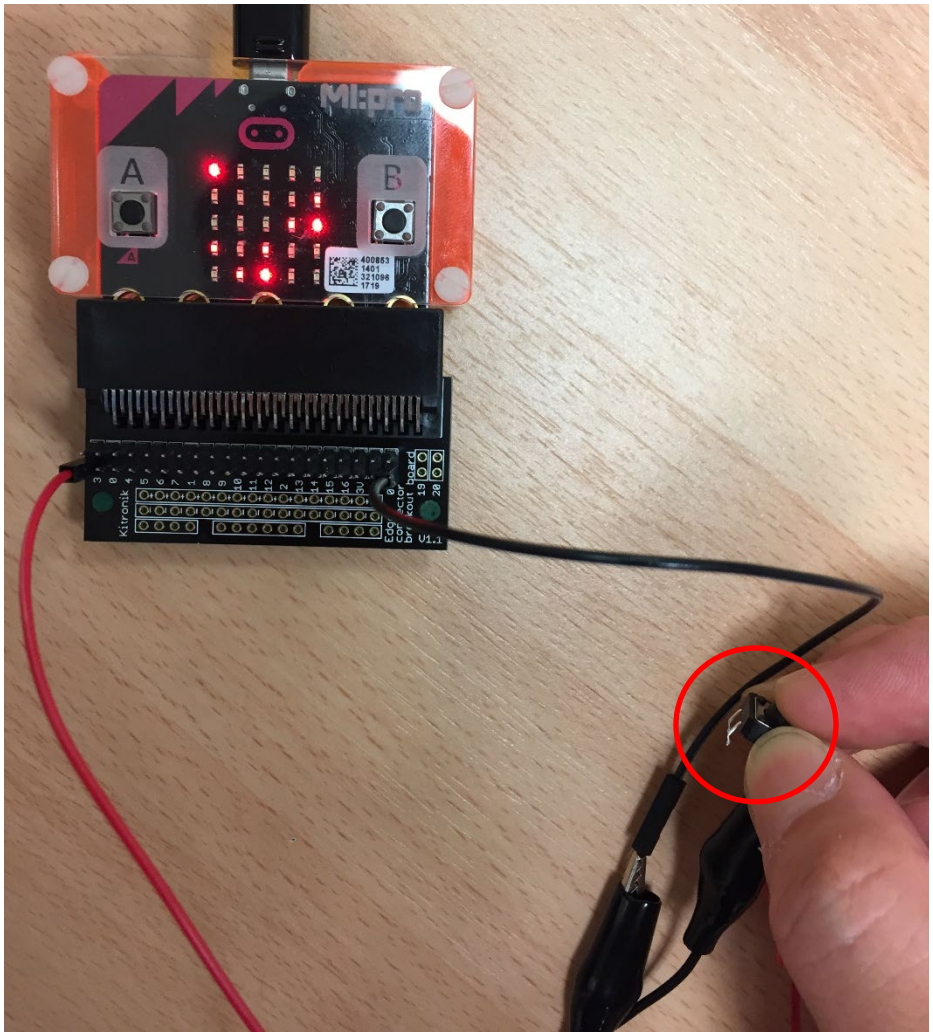
More about Micro:bit's Small Pins (cont.)

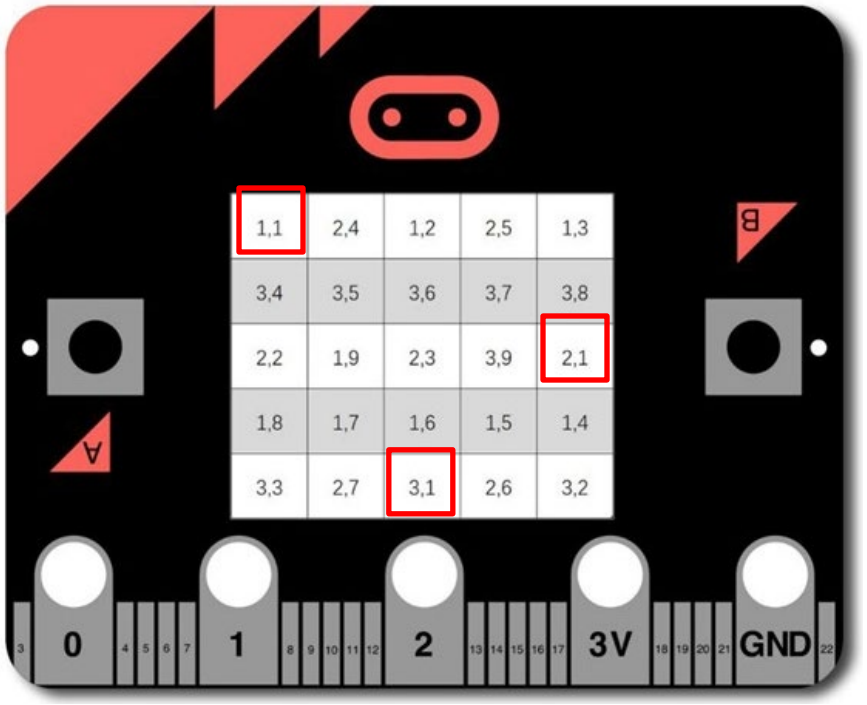
- The LED matrix is displayed as 5x5 (25) pixels (left).
- But internally, they are <u>not</u> arranged in a neat row column numbering (right):
  - Internally, the LEDs are organised into 9 columns and 3 rows.
  - Columns 8 and 9 drive only two LEDs each.
  - This arrangement makes the PCB layout easier.





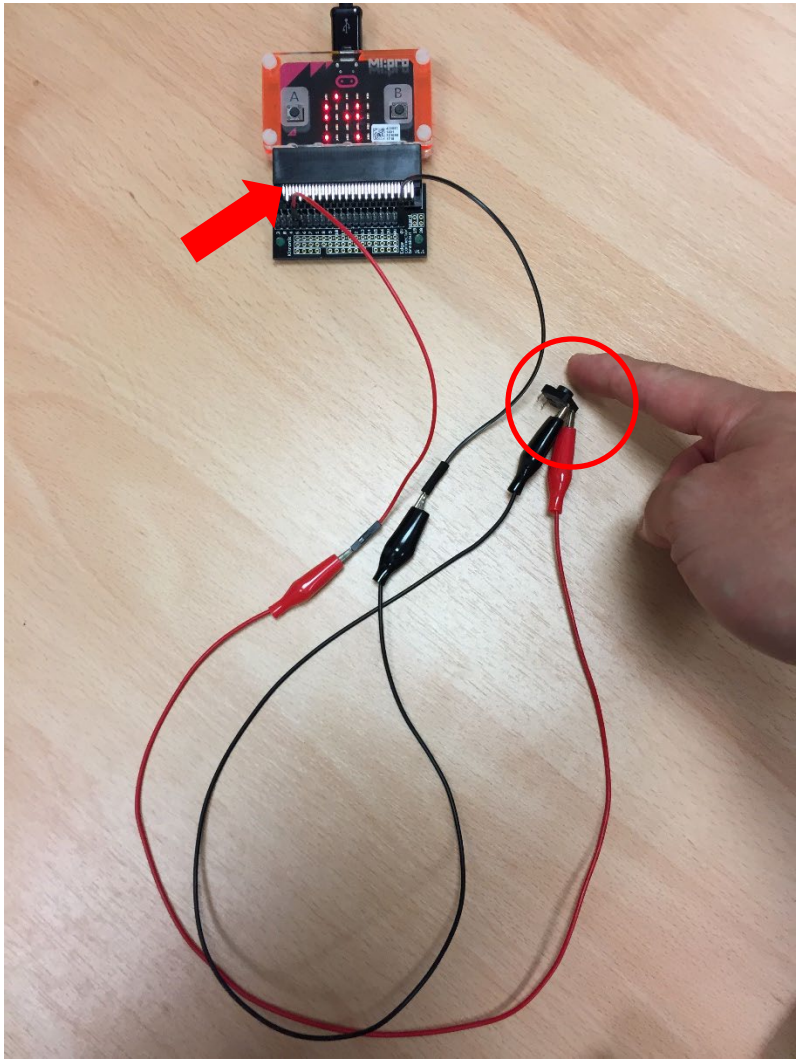
More about Micro:bit's Small Pins (cont.)





- In this demonstration, we connect a typical push button to Pin 3 and Pin 22 (GND) to drive LED Column 1 on.
- Note the peculiar pattern of the LEDs that are turned on.

More about Micro:bit's Small Pins (cont.)



- In this demonstration, we connect a typical push button to Pin 5 and Pin 22 (GND) to trigger Button A of the micro:bit device.
- Note the use of an edge connector to access the small pins.

More about Micro:bit's Small Pins (cont.)

Pin	Use
Pin 9	GPIO shared with LED Col 7 of the LED screen; can be used for digital I/O when the LED screen is turned off.
Pin 10	GPIO shared with LED Col 3 of the LED screen; can be used for ADC and digital I/O when the LED screen is turned off.
Pin 11	GPIO shared with Button B. This lets you trigger or detect a button "B" click externally.
Pin 12	Dedicated GPIO, for sending and sensing digital signals.
Pin 13	GPIO that is conventionally used for the serial clock (SCK) signal of the 3-wire Serial Peripheral Interface (SPI) bus.
Pin 14	GPIO that is conventionally used for the Master In Slave Out (MISO) signal of the SPI bus.
Pin 15	GPIO that is conventionally used for the Master Out Slave In (MOSI) signal of the SPI bus.
Pin 16	Dedicated GPIO (conventionally also used for SPI 'Chip Select' function).

More about Micro:bit's Small Pins (cont.)

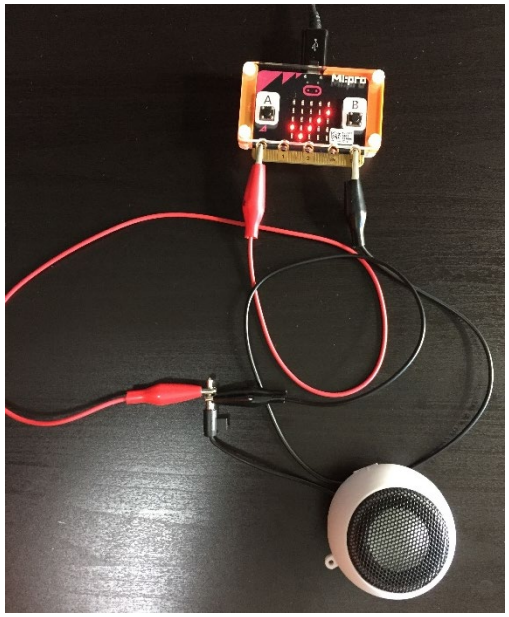
Pin	Use
Pin 17	These pins are wired to the 3V supply, like the large "3V" pad. **The precise power rating of micro:bit is 3V3 or 3.3 volts.*
Pin 18	
Pin 19	Implement the clock signal (SCL) and data line (SDA) of the I2C bus communication protocol. With I2C, several devices can be connected on the same bus and send/read messages to and from the CPU. Internally, the accelerometer and the compass are connected to I2C.
Pin 20	
Pin 21	These pins are wired to the GND pin and serve no other function.
Pin 22	

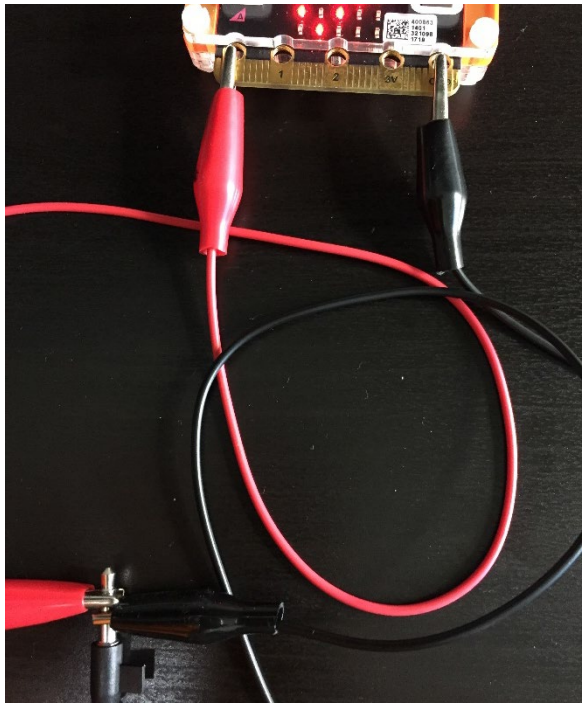
More about Micro:bit's Small Pins (cont.)

- Connecting to the small pins:
  - It is recommended that an edge connector be used to connect to the small pins, e.g., the edge connector from Kitronik.
  - We can also use a more user-friendly shield that allows peripherals to be plugged onto micro:bit, e.g., Grove Shield for micro:bit from Seeed Studio.
- We will learn how to program the GPIO pins in various ways:
  - Using a large pin directly connecting to Micro:bit's edge connector.
  - Using a small pin via an edge connector.
  - Using a shield.

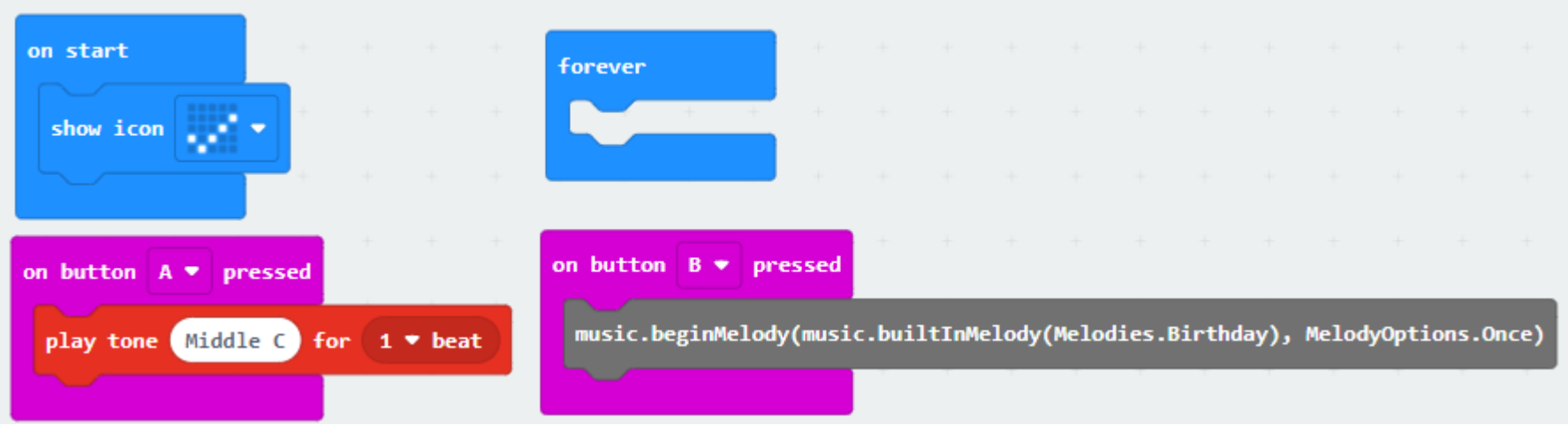
Music in Micro:bit

- Music tones are generated through Pin 0, i.e., GPIO 0.
- Micro:bit does not have an onboard speaker and so you need to attach a speaker manually with crocodile clips:
  - Attach Pin 0 to the positive input on the speaker.
  - Attach GND to the negative input.





Music in Micro:bit (cont.)



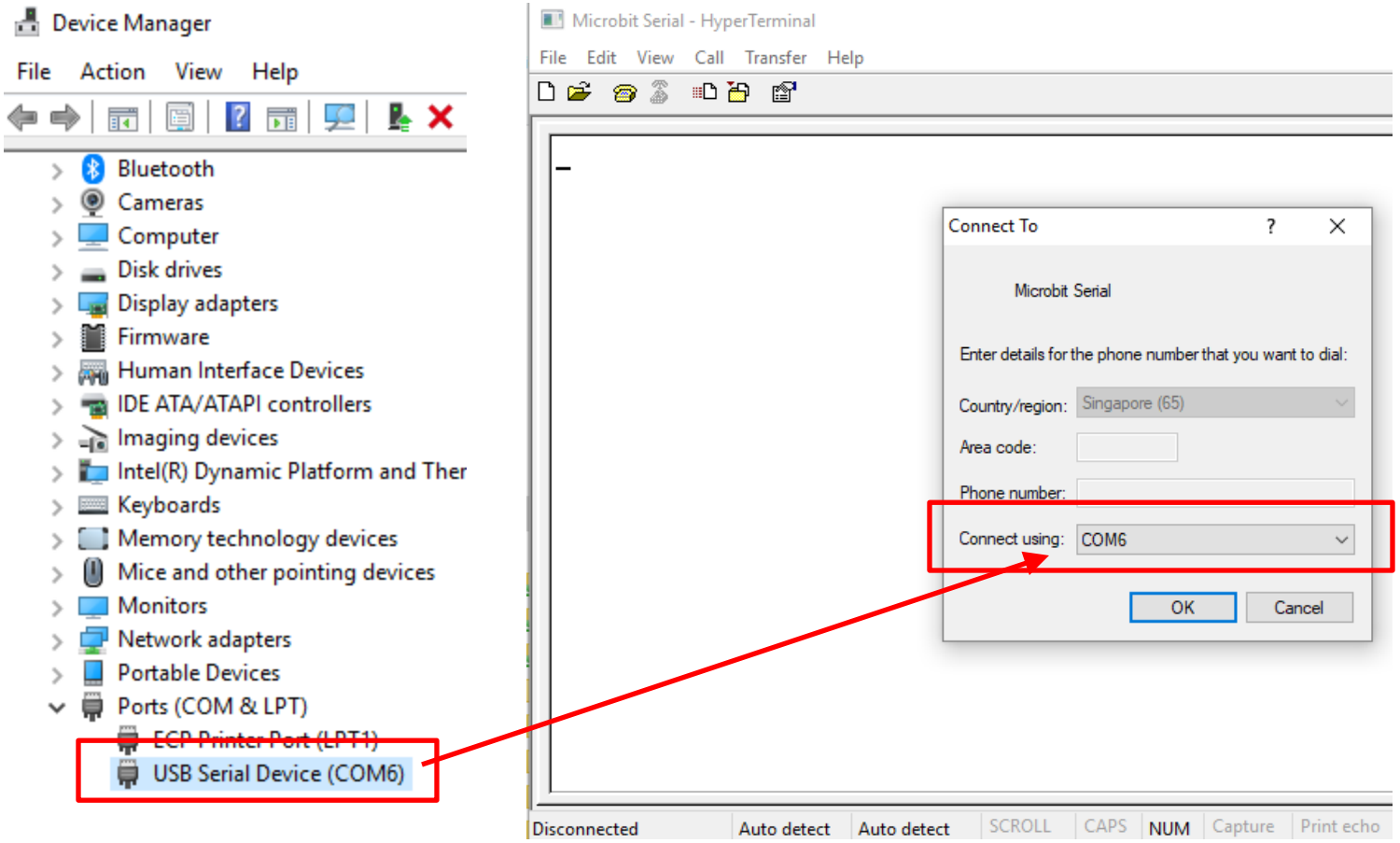

```
1  input.onButtonPressed(Button.A, function () {
2      music.playTone(262, music.beat(BeatFraction.Whole))
3  })
4  input.onButtonPressed(Button.B, function () {
5      music.beginMelody(music.builtInMelody(Melodies.Birthday), MelodyOptions.Once)
6  })
7  basic.showIcon(IconNames.Yes)
8  basic.forever(function () {
9
10 })
```

src02.js

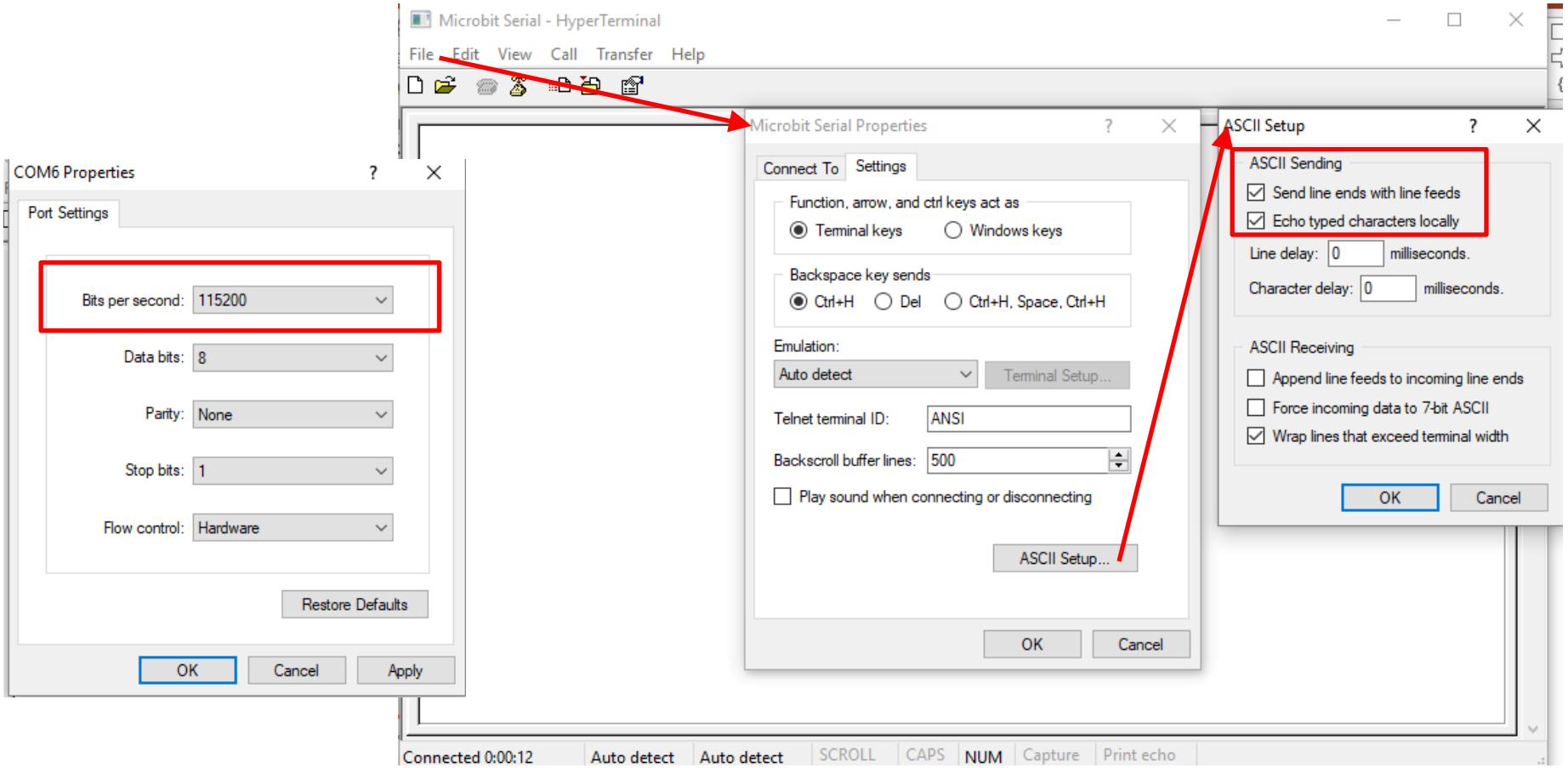
Serial Communication

- Micro:bit supports reading and writing data over a serial connection.
- By default, micro:bit is set to use the USB connection for serial data:
  - When micro:bit is connected to your computer, it will appear as a new "USB Serial Device" to Windows.
  - You need to know the COM port number and use a suitable terminal emulator program such as HyperTerminal to communicate with the micro:bit.
  - In the screenshot on the next slide, micro:bit is assigned COM6.
  - The baud rate should be set to 115200.

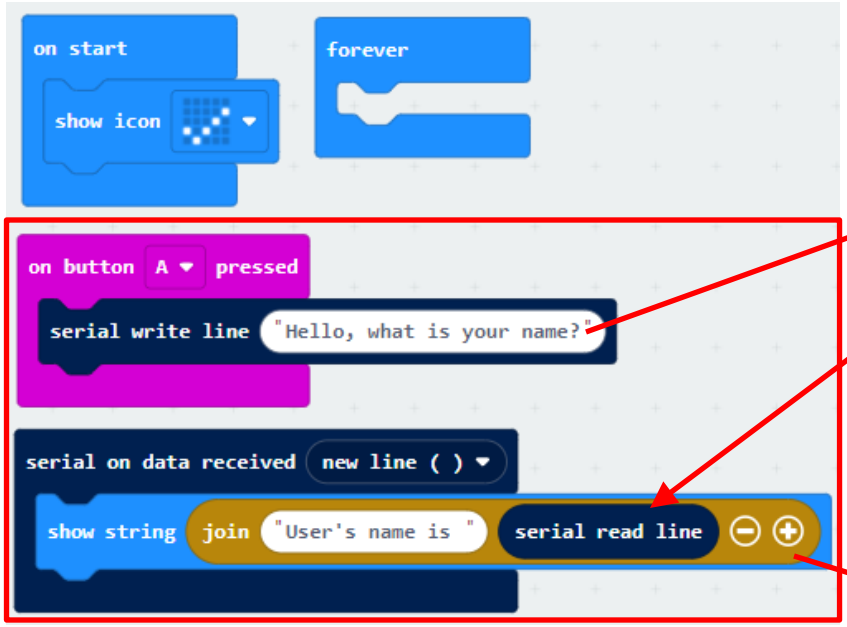
Serial Communication (cont.)

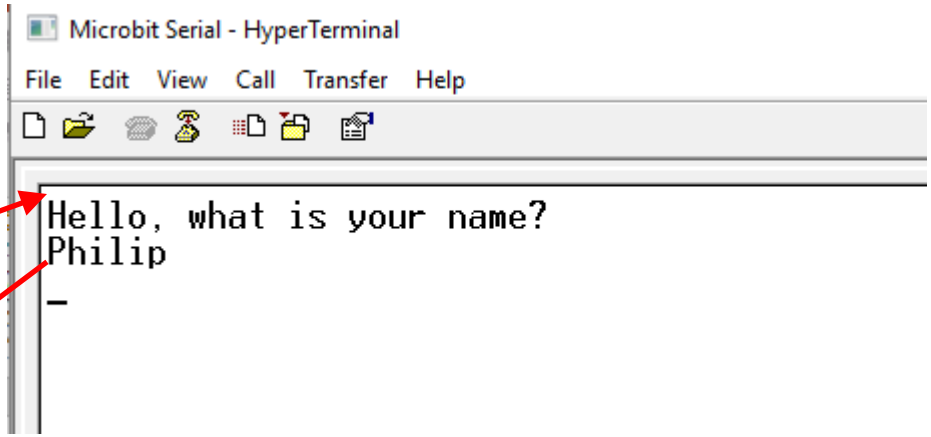


Serial Communication (cont.)



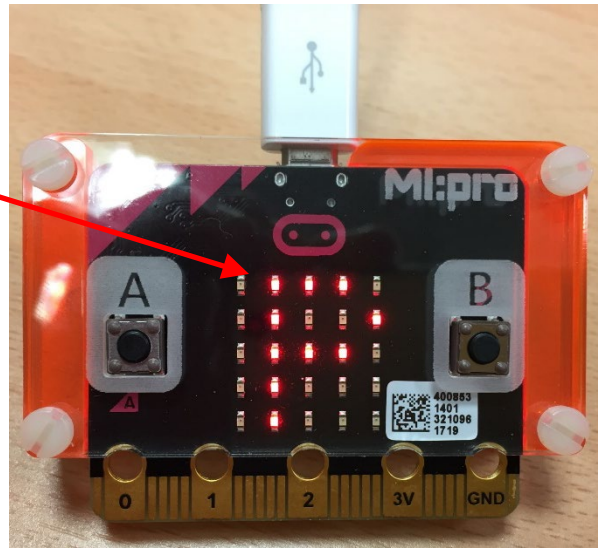
Serial Communication (cont.)






```
1  input.onButtonPressed(Button.A, function () {
2      serial.writeLine("Hello, what is your name?")
3  })
4  serial.onDataReceived(serial.delimiters(Delimiters.NewLine), function () {
5      basic.showString("User's name is " + serial.readLine())
6  })
7  basic.showIcon(IconNames.Yes)
8  basic.forever(function () {
9
10 })
```

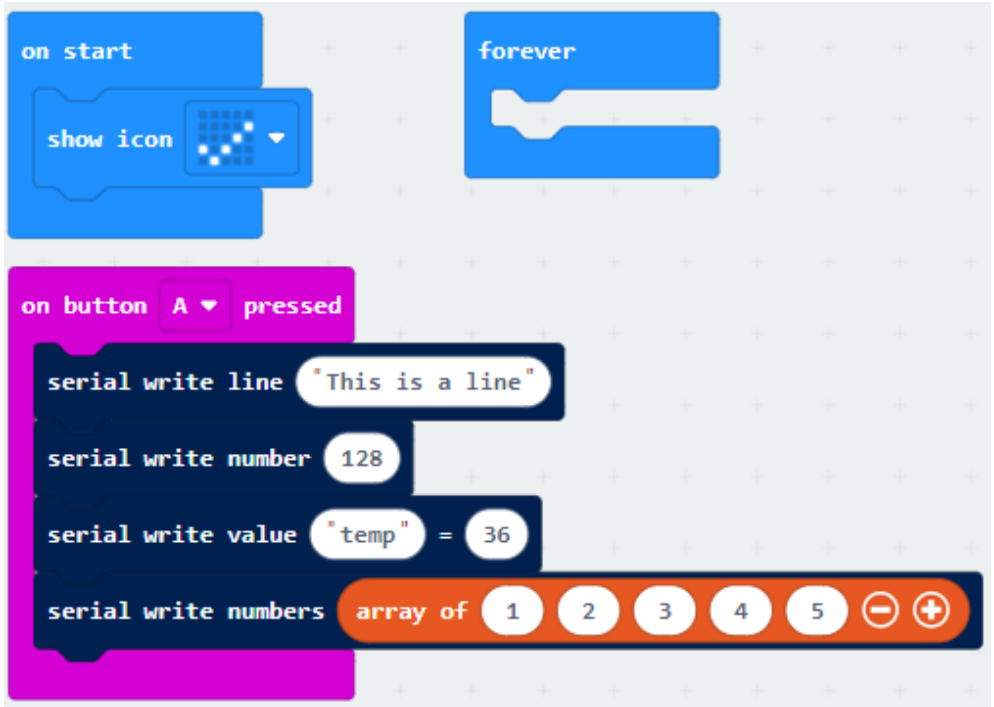
src03.js



Serial Communication (cont.)

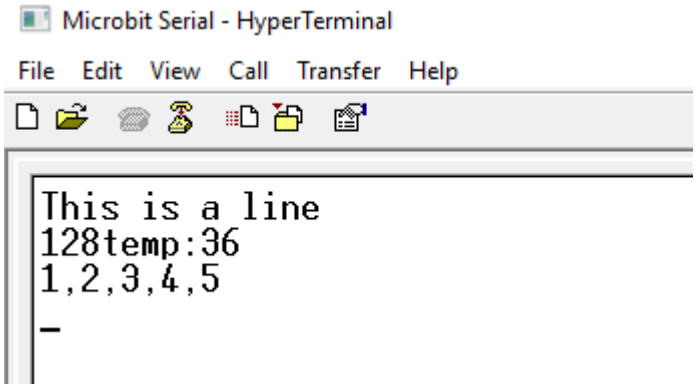
- Various types of data value can be sent over serial communication:
  - `writeLine` – A line of text.
  - `writeNumber` – A numeric value.
  - `writeValue` – A name-value pair as a line.
  - `writeString` – A piece of text.
  - `writeNumbers` – An array of numeric values as CSV (comma separated value).
- We can run a conventional program on the computer to listen on the serial port and receive data for further processing, e.g., relay to a server-side backend.

Serial Communication (cont.)




```
1  input.onButtonPressed(Button.A, function () {
2      serial.writeLine("This is a line")
3      serial.writeNumber(128)
4      serial.writeValue("temp", 36)
5      serial.writeNumbers([1, 2, 3, 4, 5])
6  })
7  basic.showIcon(IconNames.Yes)
8  basic.forever(function () {
9
10 })
```

src04.js

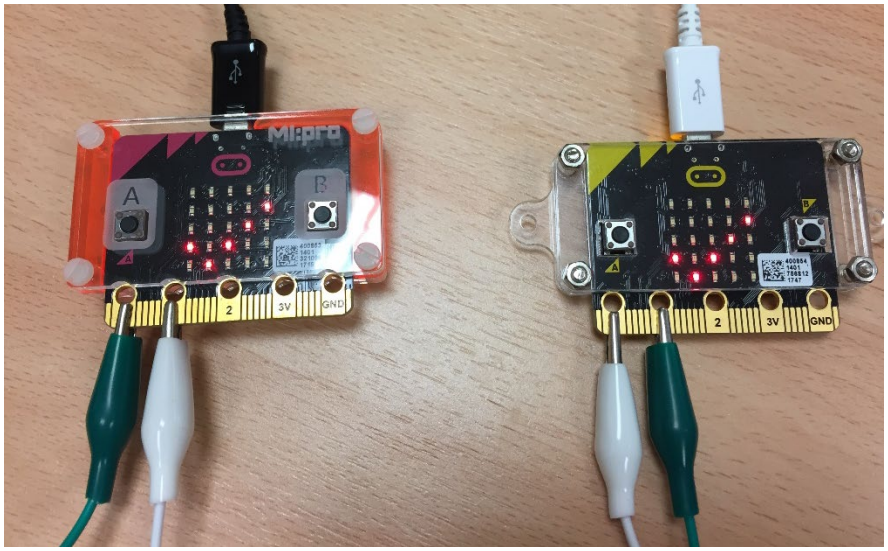


Serial Communication Over GPIO

- We can configure Micro:bit's serial communication to use the pins instead of USB.
- The `redirect` block is used to set the input and output for the serial connection to be on the pins:
  - `tx` – The transmit pin to send serial data on.
  - `rx` – The receive pin to receive serial data on.
  - `rate` – The baud rate for transmitting and receiving data, e.g., from 300 to 115200.
- Can direct the serial input and output back to use the USB connection using the `redirectToUSB` block.

Serial Communication Over GPIO (cont.)

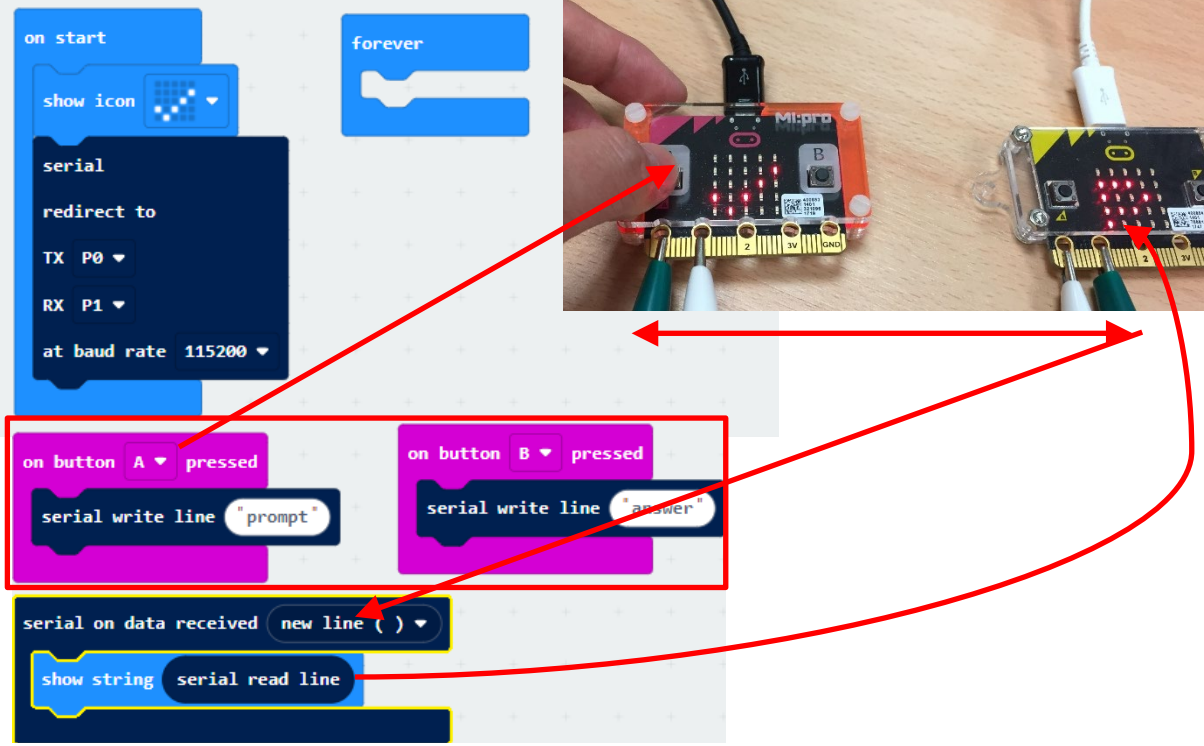
- We can enable peer-to-peer serial communication between two micro:bit devices by connecting their large GPIO pins using crocodile clips:
  - Connect the tx pin (GPIO 0) of one micro:bit to the rx pin (GPIO 1) of another micro:bit.
  - Connect the rx pin (GPIO 1) of the same micro:bit to the tx pin (GPIO 0) of the other micro:bit.
  - Both micro:bit devices will then be loaded with the same program.



Serial Communication Over GPIO (cont.)

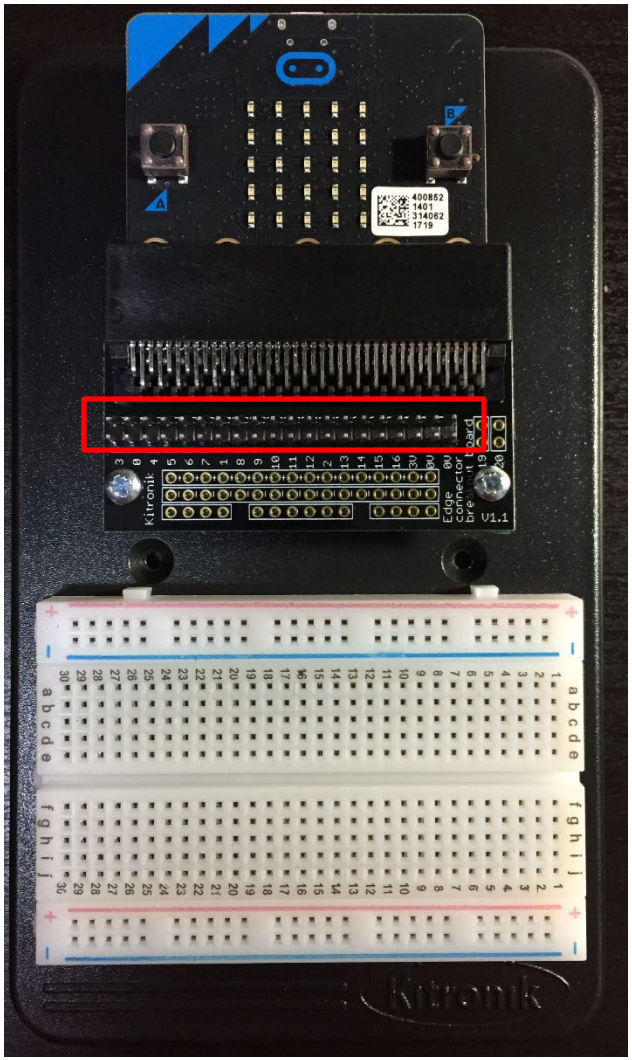
```
1  input.onButtonPressed(Button.A, function () {
2      serial.writeLine("prompt")
3  })
4  input.onButtonPressed(Button.B, function () {
5      serial.writeLine("answer")
6  })
7  serial.onDataReceived(serial.delimiters(Delimiters.NewLine), function () {
8      basic.showString(serial.readLine())
9  })
10 basic.showIcon(IconNames.Yes)
11 serial.redirect(
12 SerialPin.P0,
13 SerialPin.P1,
14 BaudRate.BaudRate115200
15 )
16 basic.forever(function () {
17
18 })
```

src05.js



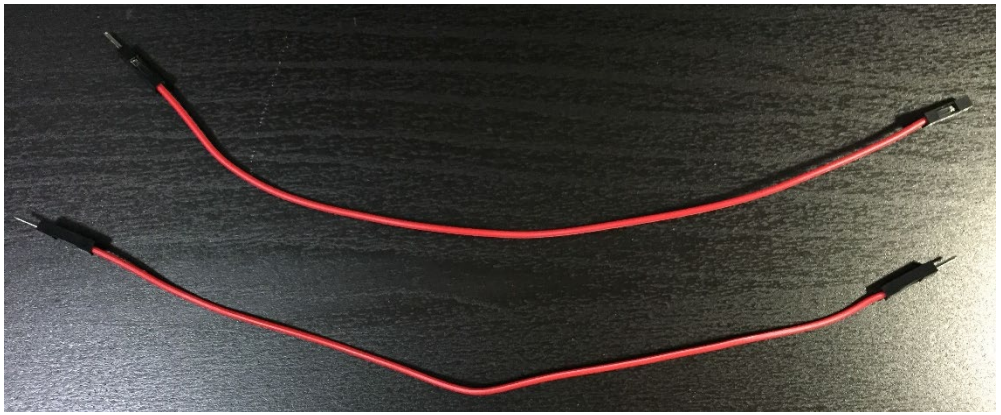
Using an Edge Connector

- The Kitronik Inventor's Kit for BBC micro:bit comes with a base plate for housing the following components:
  - 40 pins edge connector breakout board:
    - Only the lower 20 pins are used.
  - The micro:bit itself is inserted into the breakout board.
  - A half-size breadboard.



Using an Edge Connector (cont.)

- Connecting the peripherals to micro:bit's pins:
  - Male-female jumpers are used to connect peripherals on the breadboard to the pins on the edge connector.
  - Male-male jumpers are used to connect peripherals across the breadboard.



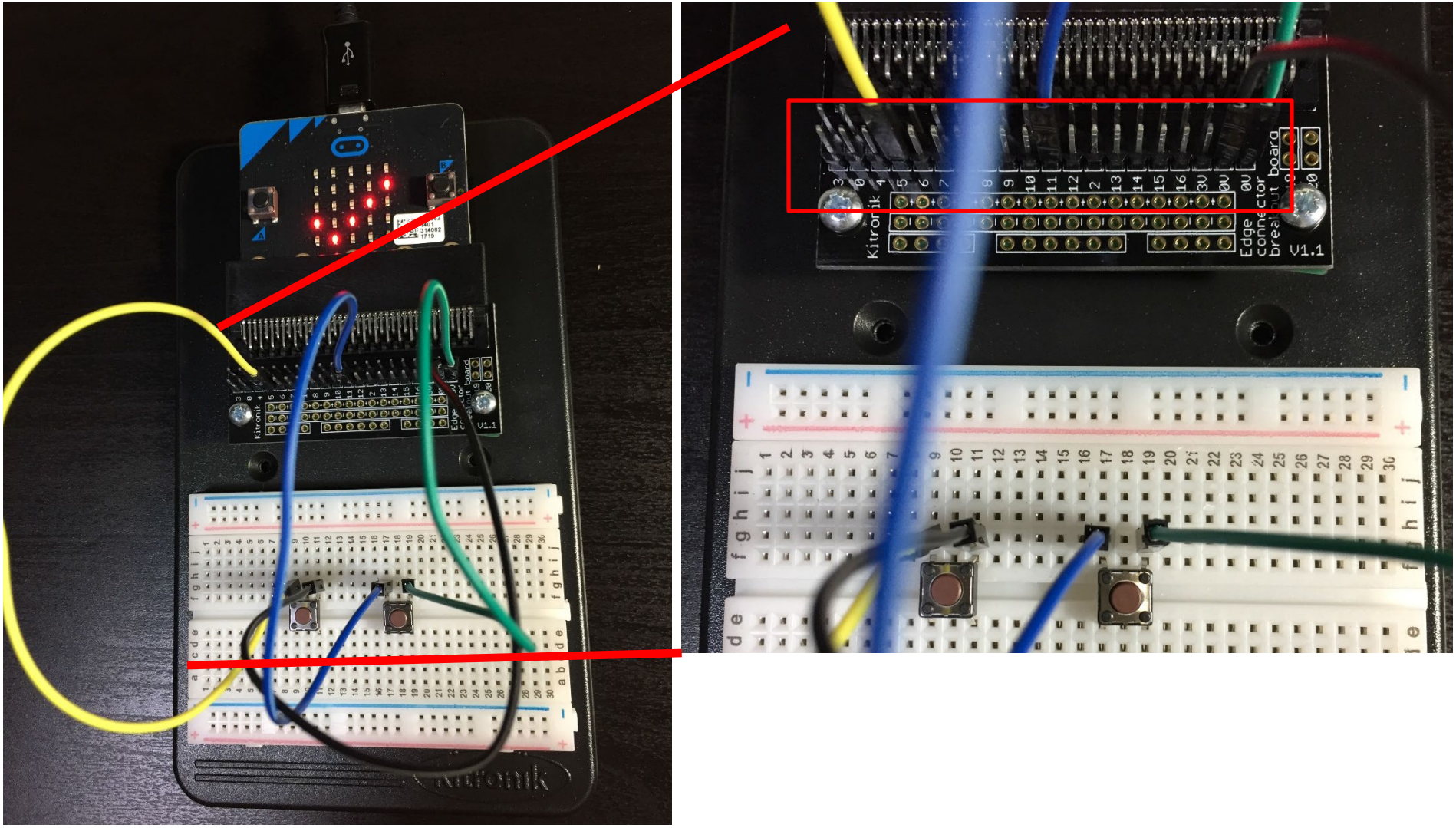
Male-female (top) and male-male (bottom) jumpers

- More details when we discuss breadboarding with the Raspberry Pi single-board computer.

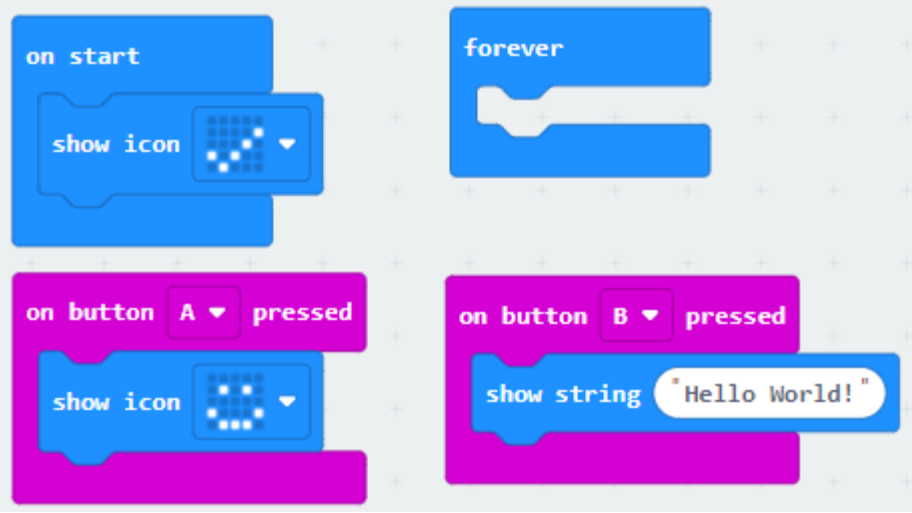
Controlling micro:bit with External Buttons

- Recall that Pin 5 and Pin 11 can be used to trigger micro:bit buttons A and B, respectively.
- We can set up a simple circuit on the breadboard as follows:
  - Connect one push button switch to Pin 5 and one of the GND pin – This will trigger button A.
  - Connect another push button switch to Pin 11 and the other GND pin – This will trigger button B.
  - Each push button switch essentially forms its own circuit, and closing it completes the circuit triggering the respective micro:bit button.

Controlling micro:bit with External Buttons (cont.)



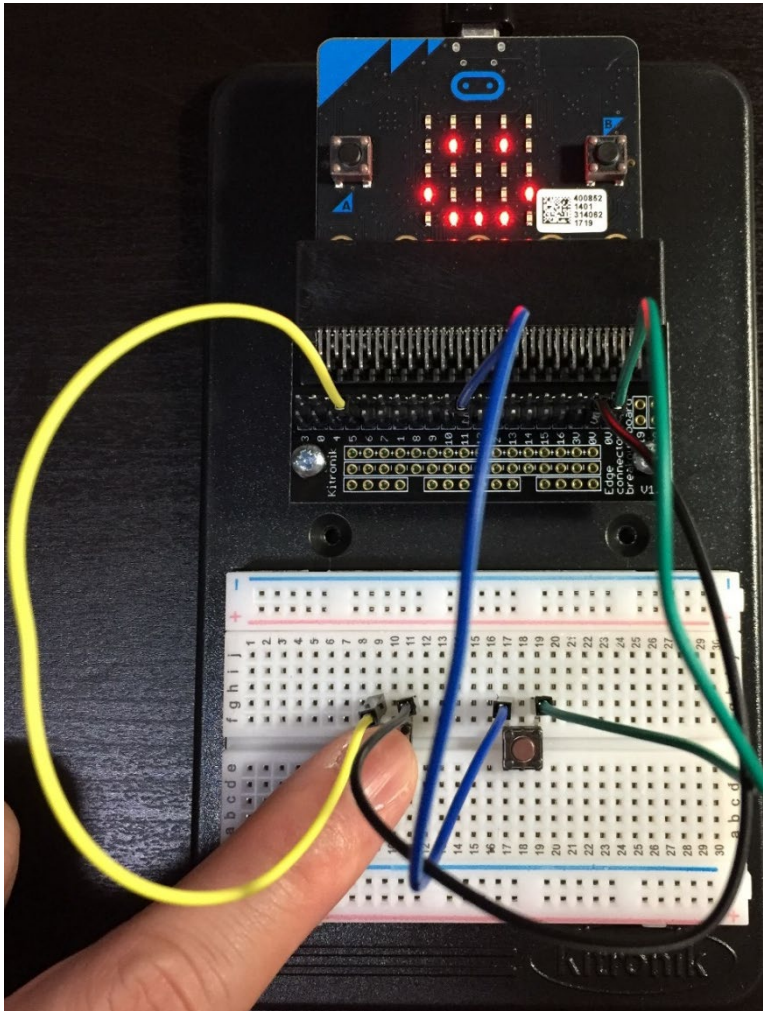
Controlling micro:bit with External Buttons (cont.)




```
input.onButtonPressed(Button.A, function () {
    basic.showIcon(IconNames.Happy)
})
input.onButtonPressed(Button.B, function () {
    basic.showString("Hello World!")
})
basic.showIcon(IconNames.Yes)
basic.forever(function () {

})
```

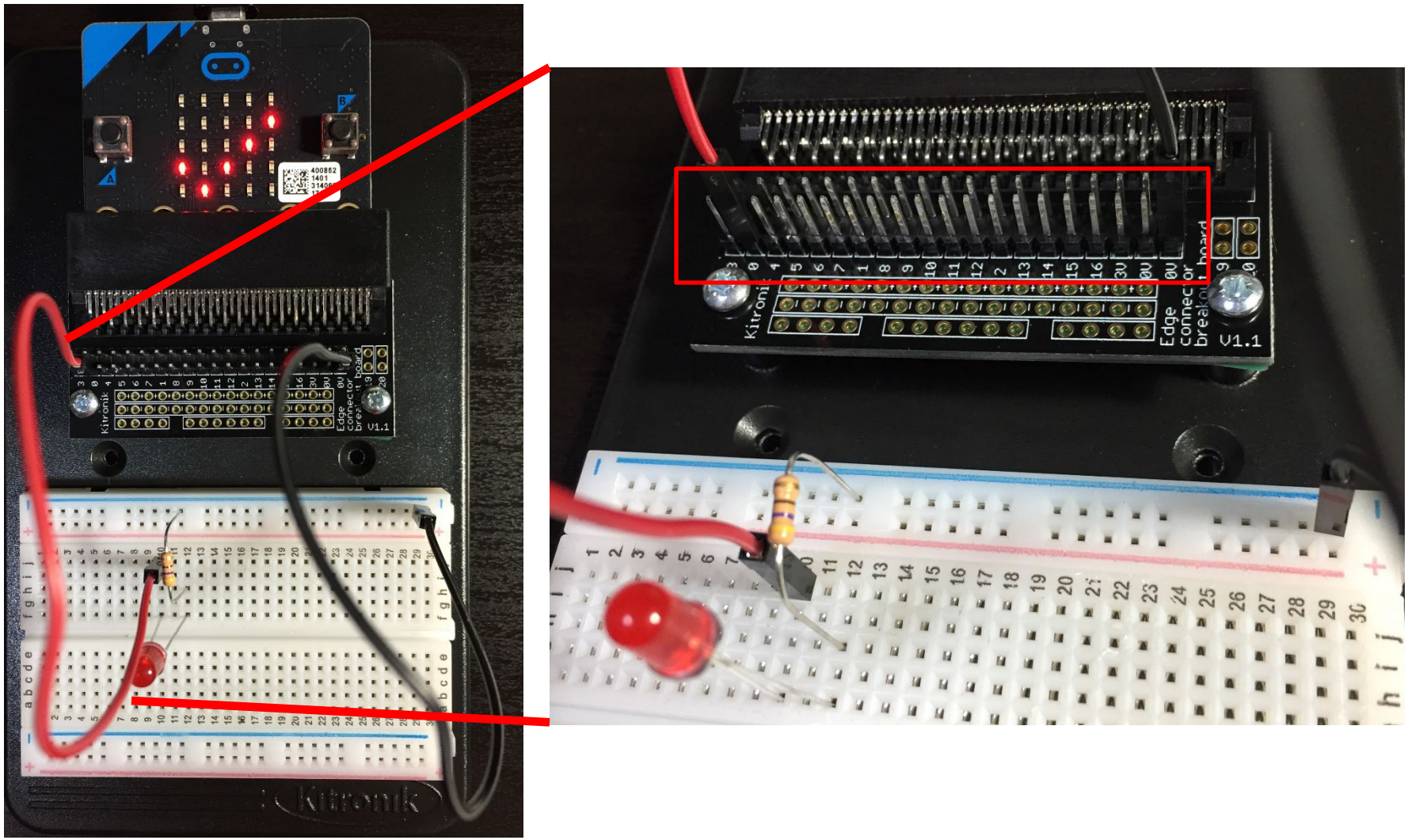
src06.js



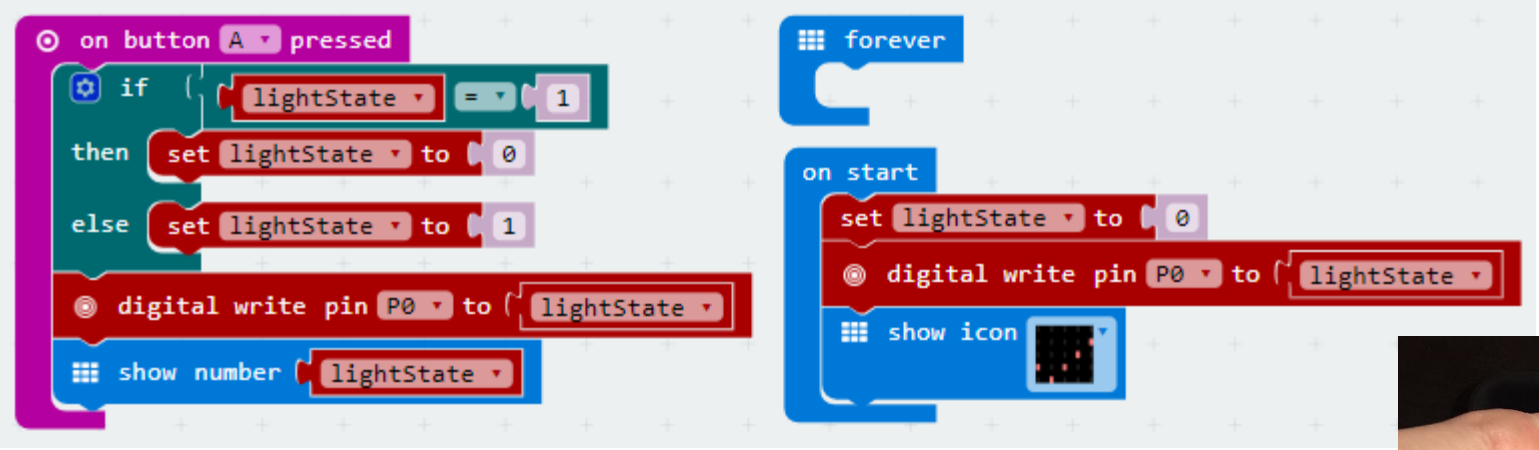
Controlling a LED

- We can setup a simple circuit on the breadboard as follow:
  - Connect GPIO Pin 0 to the longer leg of the LED.
  - Connect the shorter leg of the LED to a 47 Ohms resistor.
  - Then connect the resistor to one of the GND pin.
- We can write a program to toggle the light state of the LED as either 0 or 1 whenever the user presses button A.
- The light state is then written to GPIO Pin 0 using the `digitalWritePin` block:
  - 1 turns the LED on.
  - 0 turns the LED off.

Controlling a LED (cont.)

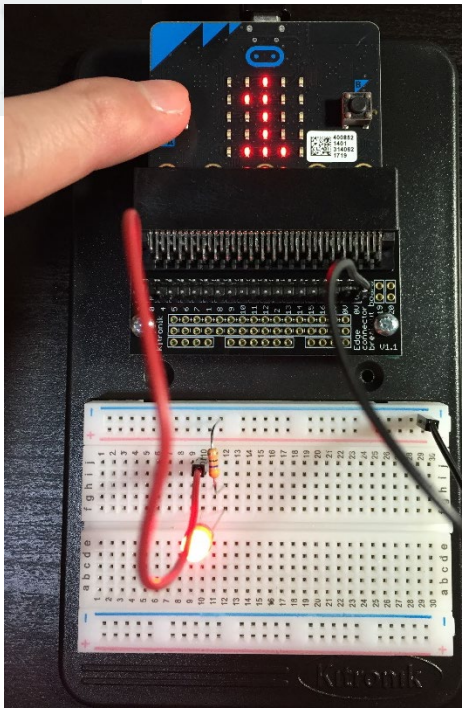


Controlling a LED (cont.)




```
let lightState = 0
input.onButtonPressed(Button.A, () => {
    if (lightState == 1) {
        lightState = 0
    } else {
        lightState = 1
    }
    pins.digitalWritePin(DigitalPin.P0, lightState)
    basic.showNumber(lightState)
})
lightState = 0
pins.digitalWritePin(DigitalPin.P0, lightState)
basic.showIcon(IconNames.Yes)
```

src07.js



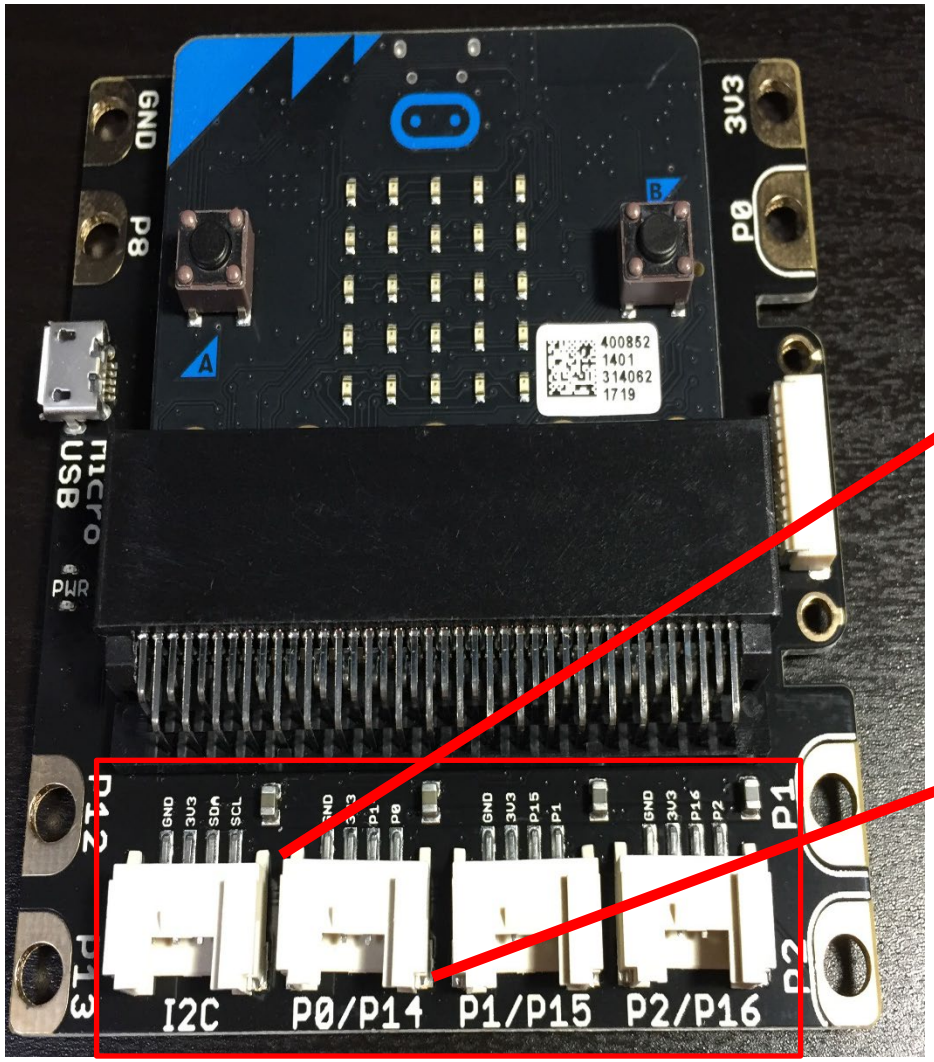
Using a Shield

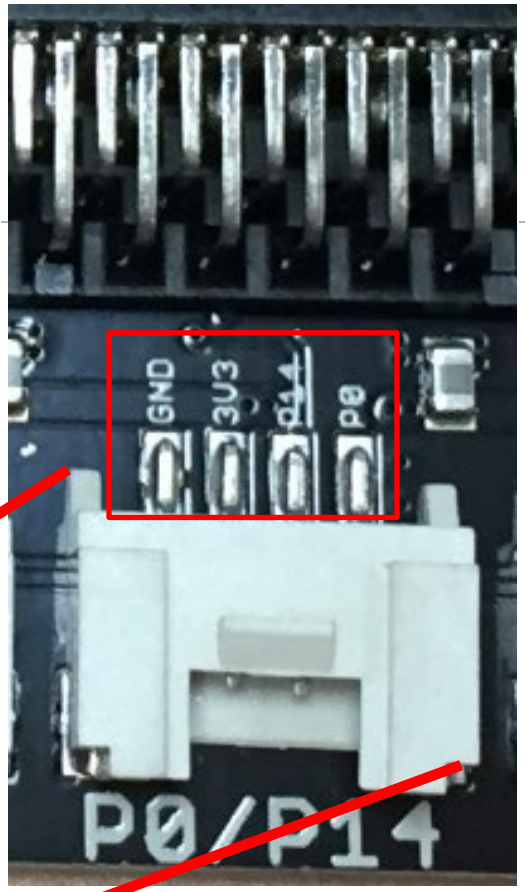
- Seeed Studio Grove modules:
  - Ready to use, building block approach to prototyping electronics.
  - Consist of a base shield and modules (sensors and other peripherals).
- micro:bit does not have Grove interface:
  - The shield provides many (four) Grove interfaces for connecting the Grove modules.
  - These interfaces map to the micro:bit's pins internally.
  - The micro:bit is plugged into the shield.
  - The Grove modules can then communicate with the micro:bit through the shield.

Using a Shield (cont.)

- To program Grove modules with micro:bit, you need to add the Grove PXT package to Microsoft MakeCode:
  - For basic modules, we can program using micro:bit's Pins blocks – GPIO Pin 0, 1 and 2.
  - For more advanced modules such as the gesture sensor and ultra sonic sensor, the Grove package provides additional blocks to easily program them.

Using a Shield (cont.)



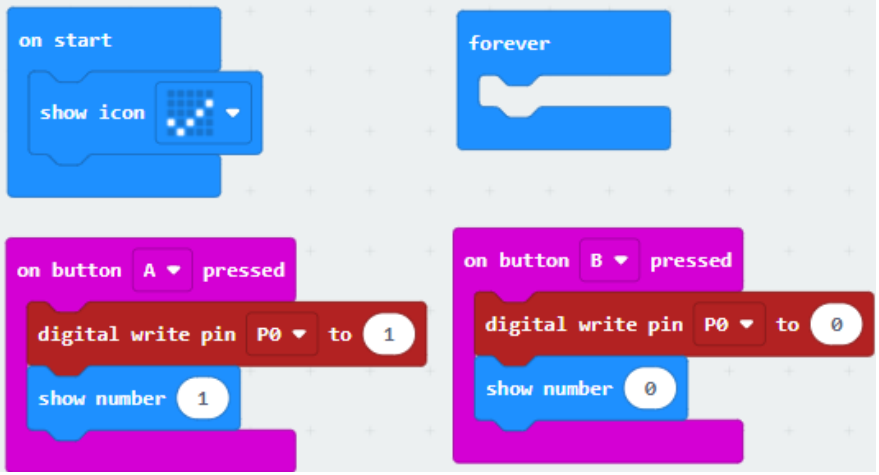


Note the use of the jumper connector. Each connector actually connects to multiple pins on the micro:bit's edge connector to form a complete circuit.

Controlling a Grove LED Module

- Grove Red LED module:
  - Monitors controls from digital ports, e.g., can be used as pilot lamp for power or signal.
  - Brightness can be adjusted by potentiometer.
  - Can be connected to either GPIO Pin 0, 1 or 2 on the shield.
- The LED module is controlled by writing to the respective GPIO Pin using the `digitalWritePin` block:
  - 1 turns the LED on.
  - 0 turns the LED off.
- This approach is similar to using the edge connector and breadboard but the Grove module is easier to connect.

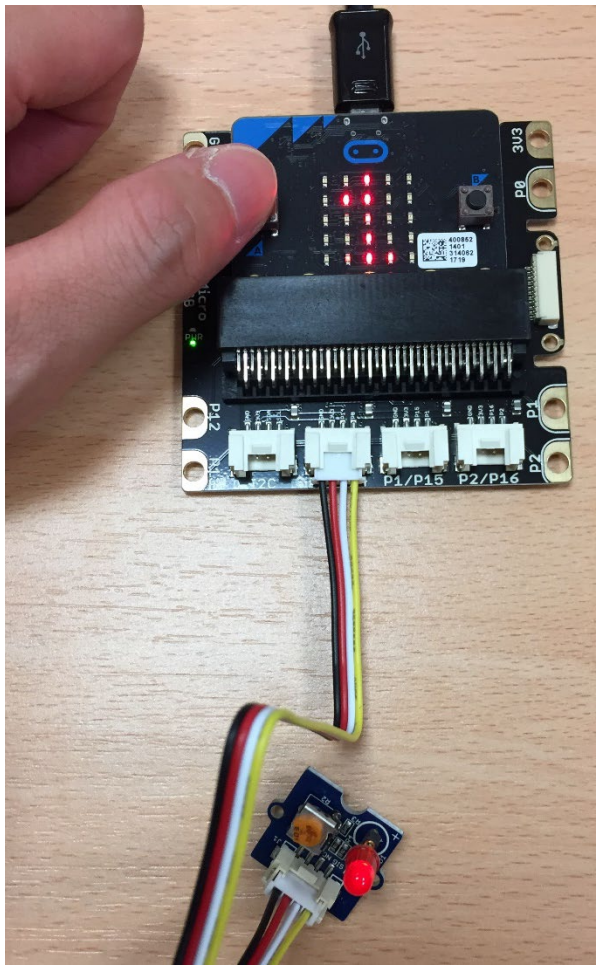
Controlling a Grove LED Module (cont.)




```
input.onButtonPressed(Button.A, function () {
    pins.digitalWritePin(DigitalPin.P0, 1)
    basic.showNumber(1)
})
input.onButtonPressed(Button.B, function () {
    pins.digitalWritePin(DigitalPin.P0, 0)
    basic.showNumber(0)
})
basic.showIcon(IconNames.Yes)
basic.forever(function () {

})
```

src08.js



Controlling a Grove LED Module (cont.)

- What other benefits does the Grove LED module offer?
  - The module comes with an onboard resistor.
  - The module comes with an onboard potentiometer for adjusting the brightness of the LED.

Analogue to Digital Converter

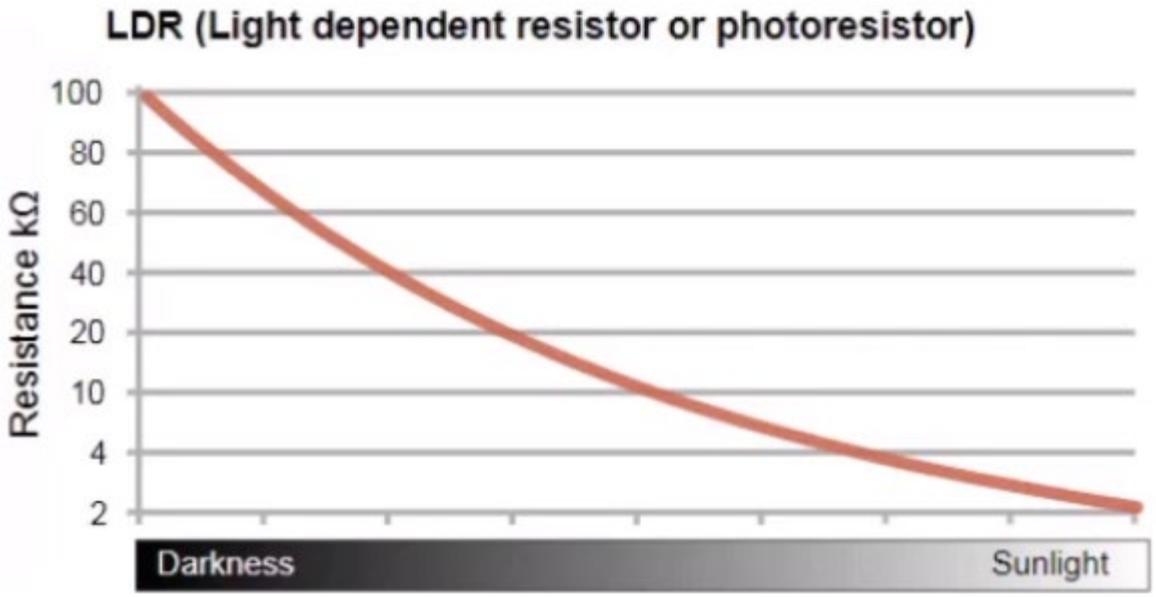
- Electronic components rely on either digital or analogue signals:
- A digital signal means that the input to or output from the component is either 3V3 or 0V:
  - Digital information is like a bit in binary – it can be one of only two values.
  - E.g., push buttons or switches are digital inputs.
  - The button is either being pressed or it is not. There is no in-between values for these two states.
- Analogue components accept input or produce output consisting of a range of voltages from 0V to 3V3.

Analogue to Digital Converter (cont.)

- micro:bit contains an **ADC (Analogue to Digital Converter)** to convert these analogue signals into a value that can be used by the microcontroller:
  - Micro:bit's ADC resolution is 10 bit.
  - It converts analogue input into a value in the range 0 – 1023 (maximum value is 2^10-1).
  - E.g., Potentiometers and light sensors rely on analogue signals.
  - Note that not all pins work with the ADC and thus you can only use certain designated pins for handling analogue signals, e.g., Pin 0 to 4.

Light Sensor

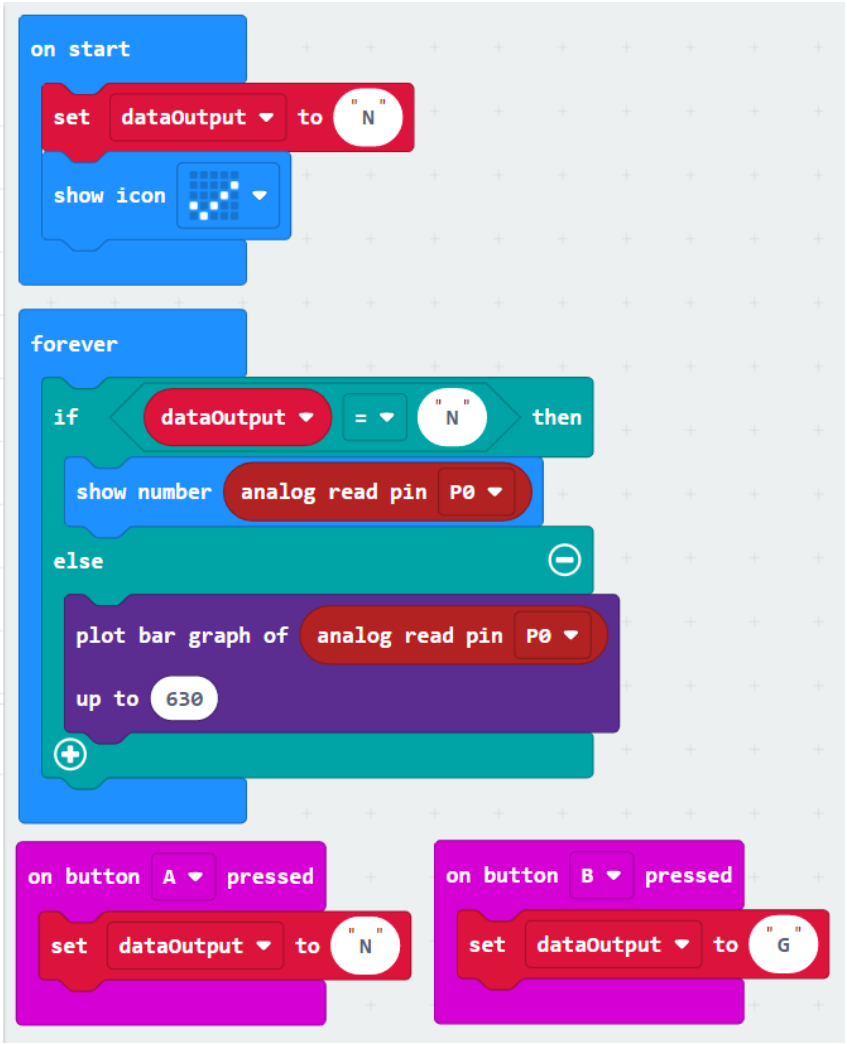
- A light sensor is essentially a light dependent resistor (LDR) or photoresistor:
  - Resistance of the LDR decreases as light intensity increases.



Light Sensor (cont.)

- Grove light sensor module:
  - Integrates a LDR with a dual operational amplifier chip LM358 that produces voltage corresponding to intensity of light (i.e., based on the LDR's resistance value).
  - The output electrical signal is analogue value, i.e., the brighter the light is, the larger the value.
  - The analogue value is converted to different numerical ranges depending on the ADC of the microcontroller.
  - For micro:bit, the output is from 0 – 630 (maximum brightness).
- The sensor value is obtained by reading from the respective GPIO Pin using the `analogReadPin` block.

Light Sensor (cont.)




```
1   input.onButtonPressed(Button.A, function () {
2       dataOutput = "N"
3   })
4
5   input.onButtonPressed(Button.B, function () {
6       dataOutput = "G"
7   })
8
9   let dataOutput = ""
10
11  dataOutput = "N"
12  basic.showIcon(IconNames.Yes)
13
14  basic.forever(function () {
15      if (dataOutput == "N") {
16          basic.showNumber(pins.analogReadPin(AnalogPin.P0))
17      } else {
18          led.plotBarGraph(
19              pins.analogReadPin(AnalogPin.P0),
20              630
21          )
22      }
23  })
```

src09.js

Light Sensor (cont.)

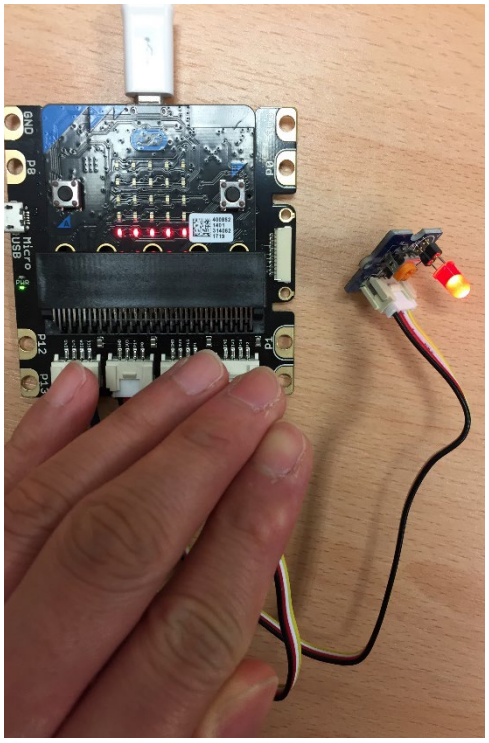
- A light sensor can be used easily to build a smart light that turns on when the ambient lighting is dimmed:
  - In the following demonstration, we turn on the Grove LED module when the light sensor value falls below certain threshold.

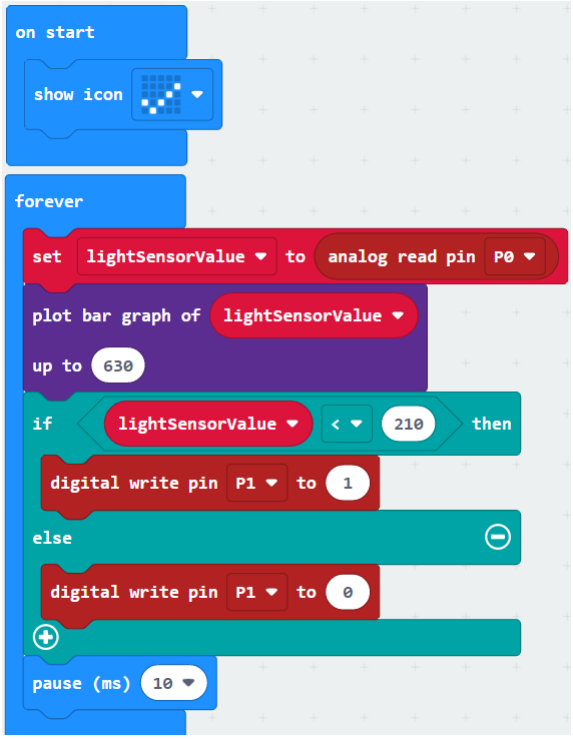
```
let lightSensorValue = 0

basic.showIcon(IconNames.Yes)

basic.forever(function () {
    lightSensorValue = pins.analogReadPin(AnalogPin.P0)
    led.plotBarGraph(
        lightSensorValue,
        630
    )
    if (lightSensorValue < 210) {
        pins.digitalWritePin(DigitalPin.P1, 1)
    } else {
        pins.digitalWritePin(DigitalPin.P1, 0)
    }
    basic.pause(10)
})
```

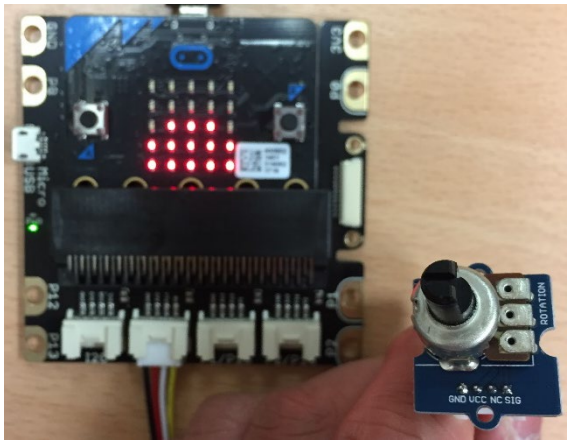
src10.js





Rotary Angle Sensor

- Grove rotary angle sensor module:
  - This is essentially a potentiometer that produces analogue output between 0 and Vcc, i.e., 3V3, for micro:bit.
  - The angular range is 300 degrees with a linear change in value.
  - The resistance value is 10k ohms.
- The sensor value is obtained by reading from the respective GPIO Pin using the `analogReadPin` block:
  - Same sample code as src09.js but with the maximum value of the `plotBarGraph` block set to 1023 (recall that micro:bit's ADC is 10 bit).



This figure shows the potentiometer roughly at the half-way mark with the graph plotted at about a value of 512

Gesture Sensor

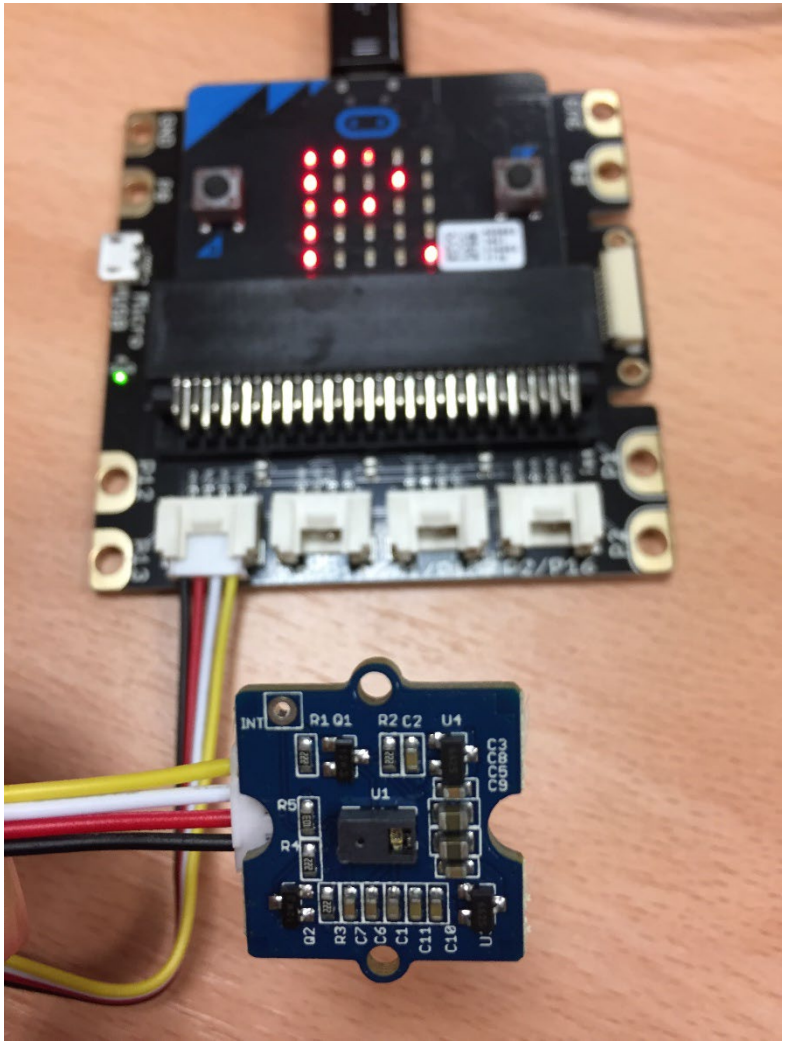
- Grove gesture sensor module:
  - Integrates gesture recognition function with general I2C interface into a single chip.
  - Micro:bit's Pin 19 and 20 are for I2C peripherals.
  - It can recognise 9 gestures with a simple swipe of the hand:
    - Move up, move down, move left, move right,, move forward and move backward.
    - Rotate clockwise and rotate anticlockwise.
    - Wave
- The sensor value is obtained using the `grove.onGesture` block:
  - This is an external package that needs to be added into the project.

Gesture Sensor (cont.)


```
1  grove.onGesture(GroveGesture.Down, function () {
2      basic.showString("D")
3  })
4  grove.onGesture(GroveGesture.Right, function () {
5      basic.showString("R")
6  })
7  grove.onGesture(GroveGesture.Backward, function () {
8      basic.showString("B")
9  })
10 grove.onGesture(GroveGesture.Up, function () {
11     basic.showString("U")
12 })
13 grove.onGesture(GroveGesture.Forward, function () {
14     basic.showString("F")
15 })
16 grove.onGesture(GroveGesture.Left, function () {
17     basic.showString("L")
18 })
19 grove.initGesture()
20 basic.showIcon(IconNames.Yes)
21 basic.forever(function () {
22
23 })
```

src11.js

Gesture Sensor (cont.)

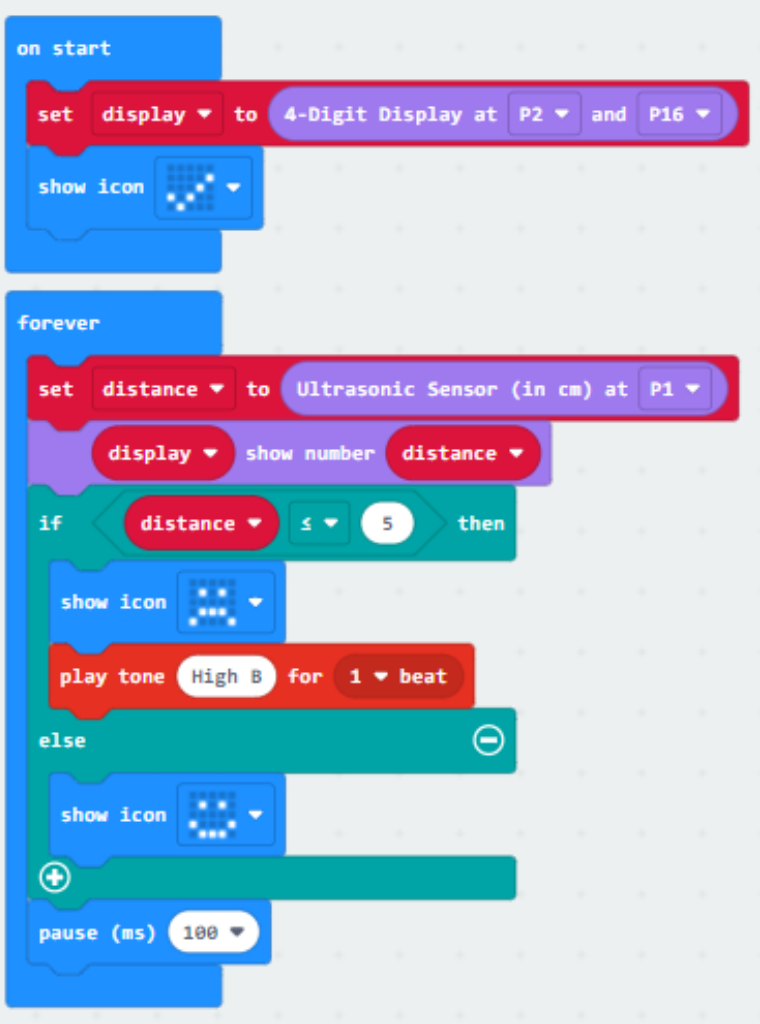


- The gesture sensor detecting a move right gesture.
- Notice that this Grove module is connected to the I2C connector instead of the usual GPIO Pin 0 connector
- See this web page for a video on the gesture sensor in action – https://www.seeedstudio.com/Grove-Gesture-PAJ7620U2.html

Ultrasonic Ranger

- Grove ultrasonic ranger module:
  - A non-contact distance measurement module.
  - Provides industrial performance.
  - Detecting range: 3cm-350cm, best in 30 degree angle.
- The sensor value is obtained using the `grove.measureInCentimeters` block.
- We can combine it with the Grove 4-digit display and Grove speaker to create a meaningful distance warning device:
  - Micro:bit will beep a warning tone and display a sad face when the ultrasonic ranger detects an object within 5 cm.

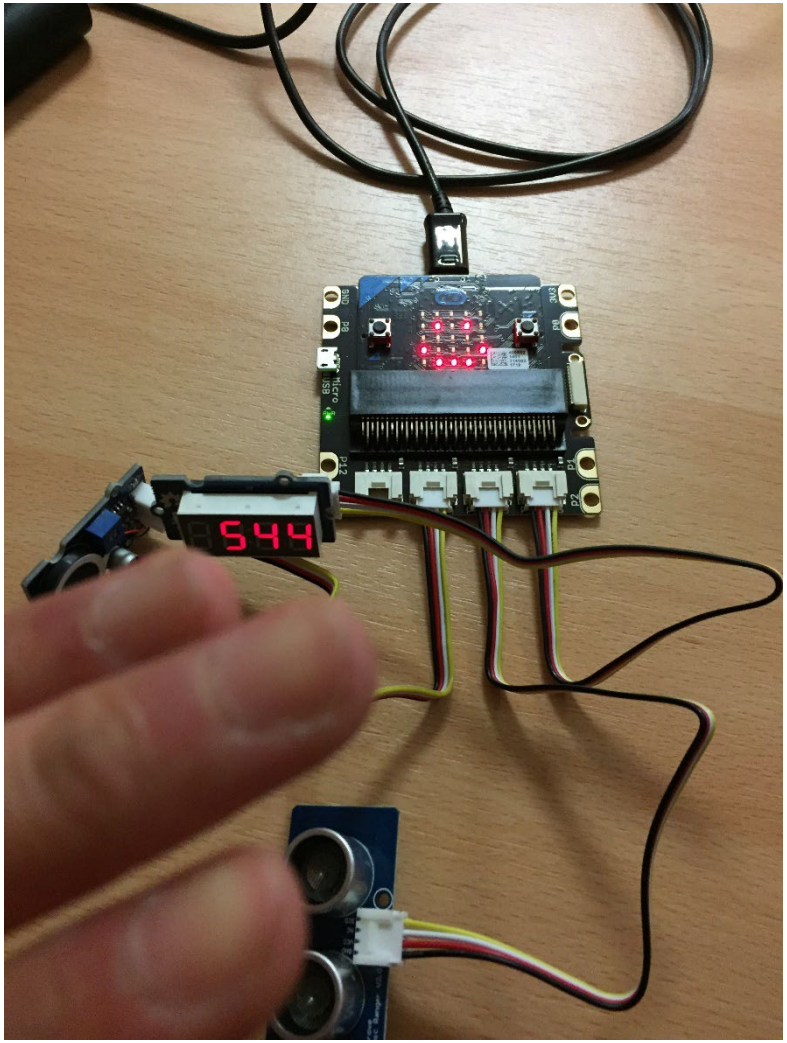
Ultrasonic Ranger (cont.)

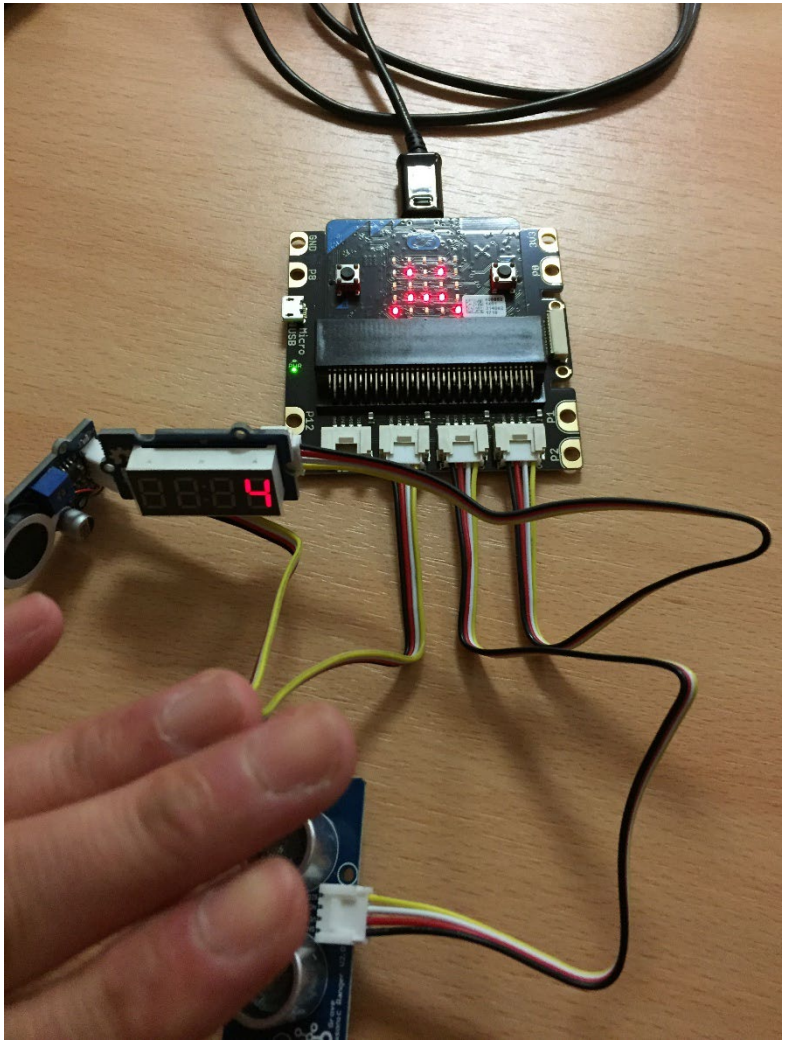



```
1  let distance = 0
2  let display = grove.createDisplay(DigitalPin.P2, DigitalPin.P16)
3  basic.showIcon(IconNames.Yes)
4  basic.forever(function () {
5      distance = grove.measureInCentimeters(DigitalPin.P1)
6      display.show(distance)
7      if (distance <= 5) {
8          basic.showIcon(IconNames.Sad)
9          music.playTone(988, music.beat(BeatFraction.Whole))
10     } else {
11         basic.showIcon(IconNames.Happy)
12     }
13     basic.pause(100)
14 })
```

src12.js

Ultrasonic Ranger (cont.)





Summary

- The 25 external connections or pins on the micro:bit device allow it to program external devices.
- Large pins can be accessed via crocodile clips or banana plugs.
- Small pins can be accessed via an edge connector or a shield.
- GPIO programming with micro:bit can involve both digital value and analogue value.
- The Grove sensor modules can greatly extend the ability of micro:bit in sensing its environment.
- Micro:bit can perform serial communication via USB or GPIO pins.

Q&A

Next Lecture…

- Learn about:
  - Overview of the Raspberry Pi single-board computer.
  - General purpose programming with Python on Raspberry Pi.
  - Basic electronic component concepts.
  - Basic Raspberry Pi GPIO programming using digital and analogue signal.