Blink a LED to send a distress message - KookaBlockly

Overview

The purpose of this sequence of resources is to allow teachers to complete most of the Stage 3 DT coding learning outcomes in just the single historical context of the electric telegraph and sending a distress signal

The Kookaberry microcontroller STEM platform programmed using the KookaBlockly visual programming editor will be used as the basis for teaching the coding components of the syllabus/curriculum.

Check out the KookaBlockly Support page for tips and instructions on using KookaBlockly and saving your apps.

You can download the SOS KookaBlockly app from within the Kbybasics zip file in the sidebar to the right. You can also download a pdf of the lesson plan from the sidebar.

Blink a LED to send a distress message - KookaBlockly

The electric telegraph

In the middle of the 18th Century, Samuel Morse co-invented the electric telegraph and the Morse Code. His code consisted of just two data states - a dot and a dash - to represent the letters of the alphabet; the numbers 0-9; and a few punctuation marks and symbols.

The electric telegraph is basically an electrical circuit between two destinations which is turned on and off by a telegraph key. The key is held down briefly to make a short signal, a dot, and slightly longer for a dash.

The electric telegraph spread around the world transforming commerce and influencing society with its instant global news. When Marconi invented long distance radio in 1901, morse code was still being used to transmit text messages. Although Samuel's code is no longer in use, the universal name for its distress signal - SOS - is still in use today.

FunFact: SOS does not mean Save Our Souls or any other message. It is just a very recognisable and easy to remember sequence of morse code letters - dot, dot, dot, dash, dash, dash, dot, dot, dot.

Age

This plan is targeted at students in years 5&6 (Stage 3) but can be introduced in Stage 3 and as a bridging course between visual and text-based programming languages in Stage 4. This is because KookaBlockly allows students to see the text-based python code behind the blocks.

Time

This plan (without extensions) will probably take 2-3 hours of classroom time if the coding activity is preceded by learning about the electric telegraph.

Learning Outcomes & Curriculum Mapping

- Learn about the invention, spread and significance of the electric telegraph. *[Electric circuits, history of communications and commerce]* ***AC9TD16P01***
- Learn about digital systems *[integration of peripherals with the Kookaberry]* ***AC9TD16K01***
- Observe how data can be represented by off and on states and by numbers and symbols *[Morse Code]* ***AC9TI6K04***
- Observe how digital systems form networks to transmit data *[MorseCode App]* ***AC9TDI6K02***
- Write and edit a programme in a visual programming language *[KookaBlockly]* ***AC9TDIP05***
- Recognise that steps in algorithms need to be accurate and precise *[Length of sleep times are critical]*
- Learn about, use, and control different output devices *[LED and Buzzer]* ***AC9TD16K01***

Blink a LED to send a distress message - KookaBlockly

- Learn how to trouble shoot through trial and error whilst programming in real time *[Vary sleep times]*
- Learn how to use branching statements (if/do) in programmes *[If button pressed]* ***AC9DI6P02***
- Learn about repeat loops *[Repeat a dot or a dash three times for the letters S and O]* ***AC9TDI6P05***
- Learn how functions can reduce the number of coding steps *[Set up functions for each of S and O]* ***AC9TDI6P05***
- Learn how to position and write text on a screen *[Vary position and size of font on the Kookaberry's screen]* ***AC9TDI6P03 AC9TDI6P05***
- Learn how to input and change user data in a programme *[Pin numbers and sleep duration]* ***AC9TDI6P04***
- Save and rename files across multiple locations *[Add a .kby suffix]*

Australian Curriculum YEARS 5 and 6											
Digital Technologies											
AC9TDI6K				**AC9TDI6P**							
01	**02**	**03**	**04**	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**
●	●		●	●	●	●		●			

Blink a LED to send a distress message - KookaBlockly

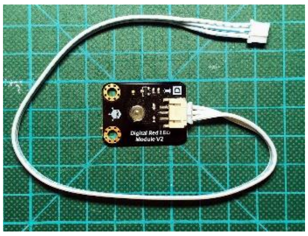
Learning Extensions

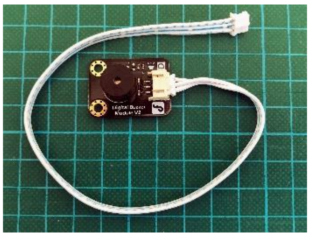
1. The SOS app shown does not have the correct dot, dash, and space interval ratios according to this Morse Code table. Adjust the sleep times to make it correct.
2. Use loops to avoid having to code every letter and sleep interval.

Resources

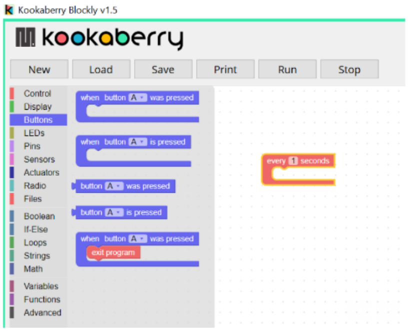
Kookaberry & USB lead

LED Module or Buzzer Module plus 3pin JST peripheral lead





KookaBlockly Visual Editor



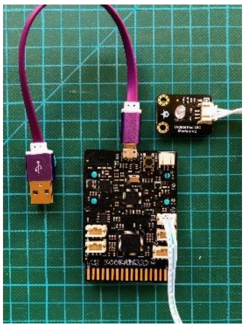
```
when button A was pressed

when button A is pressed

button A was pressed

button A is pressed

when button A was pressed
    exit program

every 1 seconds
```

MorseCode App

Generate and exchange your own Morse messages using the MorseCode app

Blink a LED to send a distress message - KookaBlockly

Connecting up



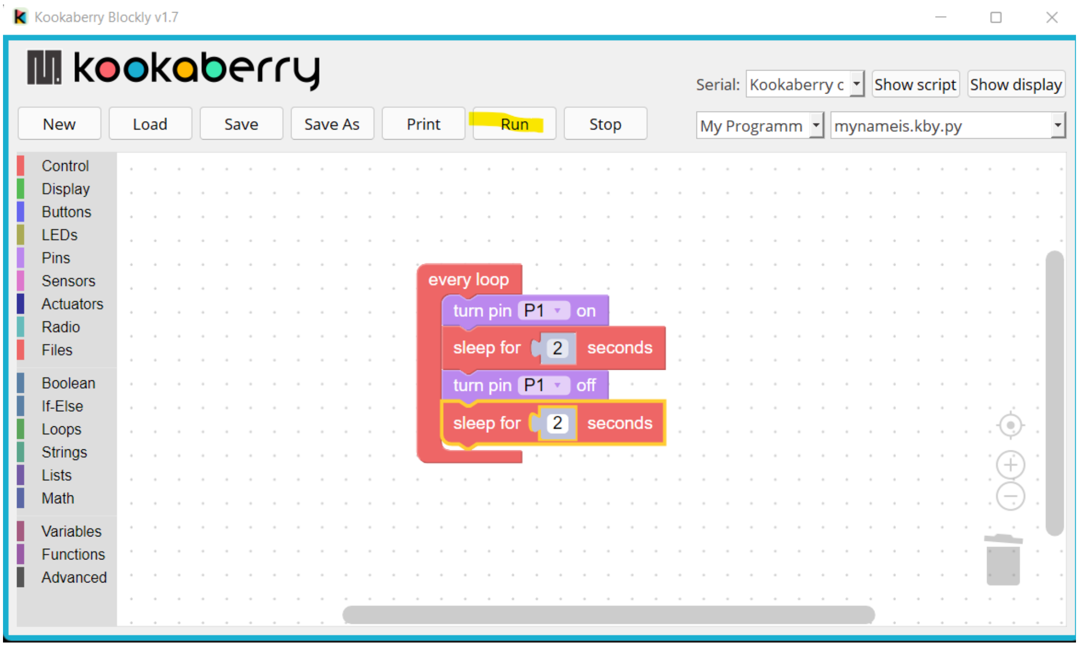
Connect the Kookaberry to your PC or Mac, and plug the LED Module into socket P1 on your Kookaberry

Blinking an LED

Open KookaBlockly; open the Control menu; and drag the "every loop" Control block onto the canvas. Then go the Pins menu and drag the "turn pin....on" into the control loop. The default Pin number is P1, but this can be set to other Pins from the drop down menu within the block.

Now go back to the Control menu and drag the "sleep for ....seconds" into the control loop. Set the sleep time for the amount of delay required (2 secs in the example shown)

Now turn P1 off for the required number of seconds by dragging the blocks into the control loop and **Run** the programme.



```
every loop
    turn pin P1 on
    sleep for 2 seconds
    turn pin P1 off
    sleep for 2 seconds
```

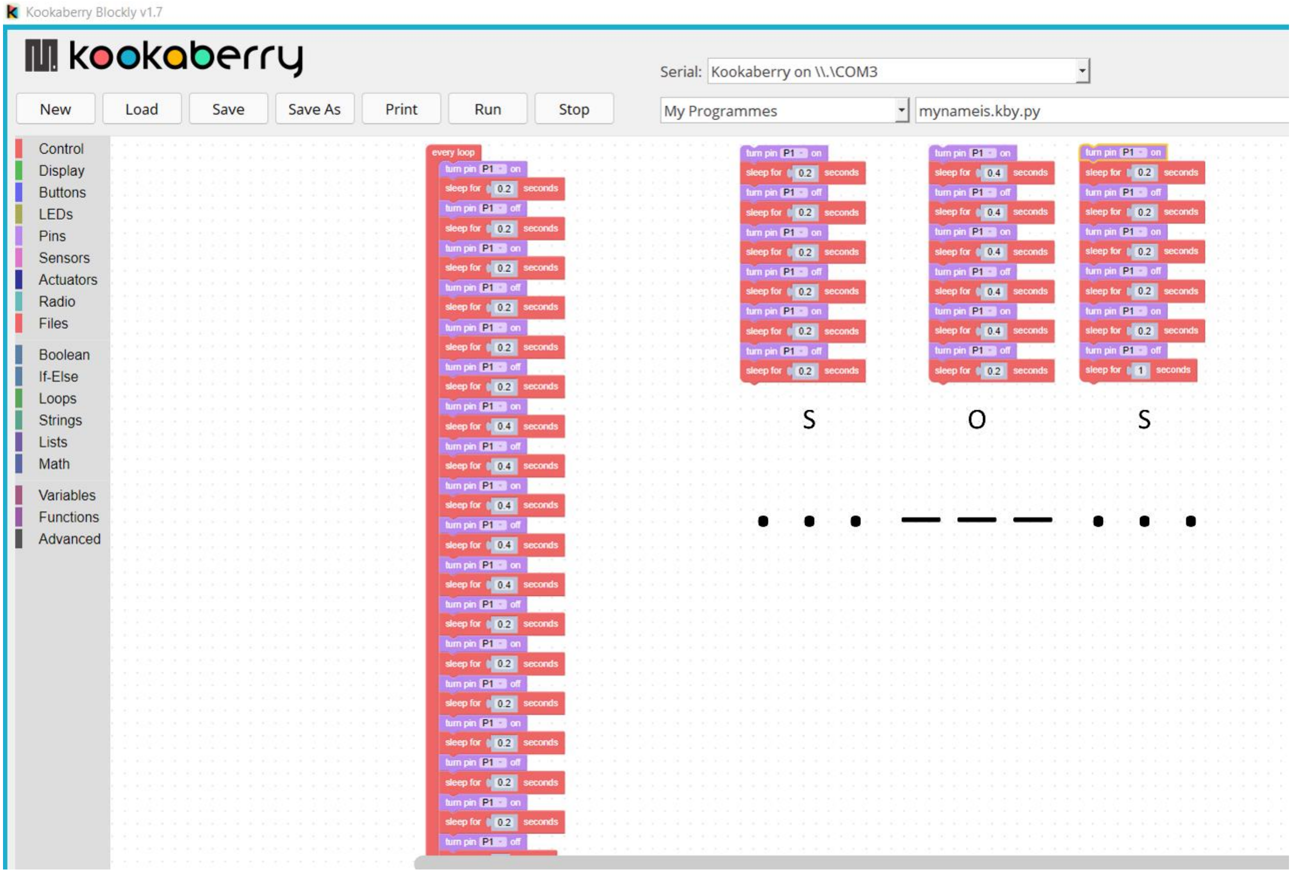
Sending a Distress Signal

Now programme the Kookaberry to send the international distress signal (SOS) in Morse Code. "S" is three dots and "O" is three dashes.

Blink a LED to send a distress message - KookaBlockly

Pick a duration time of 0.2 secs for a dot and 0.4 secs for a dash; with 1 sec interval between letters.

[Tip: Use the Duplicate function in the block dropdown edit menu. Right-click on block to see menu]



Serial: Kookaberry on \\.\COM3

My Programmes

mynameis.kby.py

```
every loop
    turn pin P1 on
    sleep for 0.2 seconds
    turn pin P1 off
    sleep for 0.2 seconds
    turn pin P1 on
    sleep for 0.2 seconds
    turn pin P1 off
    sleep for 0.2 seconds
    turn pin P1 on
    sleep for 0.2 seconds
    turn pin P1 off
    sleep for 0.2 seconds
    turn pin P1 on
    sleep for 0.4 seconds
    turn pin P1 off
    sleep for 0.4 seconds
    turn pin P1 on
    sleep for 0.4 seconds
    turn pin P1 off
    sleep for 0.4 seconds
    turn pin P1 on
    sleep for 0.4 seconds
    turn pin P1 off
    sleep for 0.2 seconds
    turn pin P1 on
    sleep for 0.2 seconds
    turn pin P1 off
    sleep for 0.2 seconds
    turn pin P1 on
    sleep for 0.2 seconds
    turn pin P1 off
    sleep for 0.2 seconds
    turn pin P1 on
    sleep for 0.2 seconds
    turn pin P1 off
```

```
turn pin P1 on
sleep for 0.2 seconds
turn pin P1 off
sleep for 0.2 seconds
turn pin P1 on
sleep for 0.2 seconds
turn pin P1 off
sleep for 0.2 seconds
turn pin P1 on
sleep for 0.2 seconds
turn pin P1 off
sleep for 0.2 seconds
```

```
turn pin P1 on
sleep for 0.4 seconds
turn pin P1 off
sleep for 0.4 seconds
turn pin P1 on
sleep for 0.4 seconds
turn pin P1 off
sleep for 0.4 seconds
turn pin P1 on
sleep for 0.4 seconds
turn pin P1 off
sleep for 0.2 seconds
```

```
turn pin P1 on
sleep for 0.2 seconds
turn pin P1 off
sleep for 0.2 seconds
turn pin P1 on
sleep for 0.2 seconds
turn pin P1 off
sleep for 0.2 seconds
turn pin P1 on
sleep for 0.2 seconds
turn pin P1 off
sleep for 1 seconds
```

.....Now hear it

Press the reset button on the back of the Kookaberry and swap the LED module for a Buzzer Module to hear the distress signal the same way as radio operators did in the past. If you have a problem, disconnect and reconnect the Kookaberry to your computer

Save it on your Kookaberry

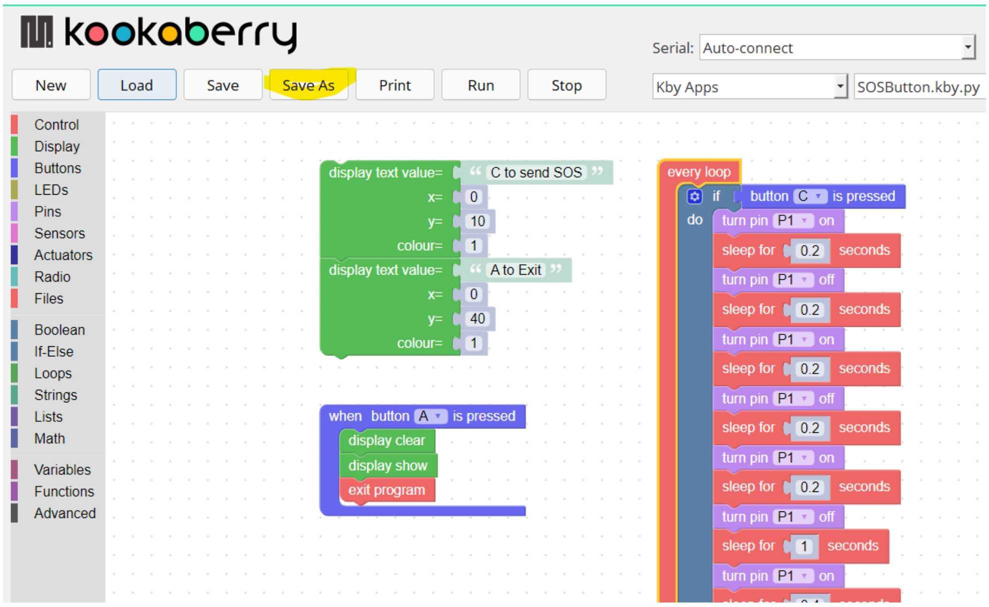
To save this file on your Kookaberry so that you can run it independently from your computer, you have to add

- an Exit programme instruction (press Button A to Exit)
- a command instruction (press button * to send)

Blink a LED to send a distress message - KookaBlockly

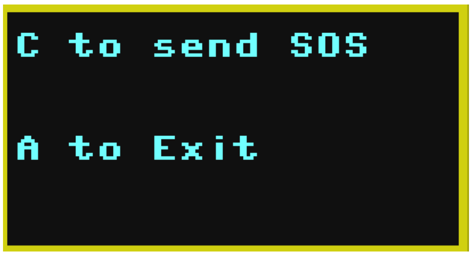
- screen prompts (what button does what...)
- a suffix .kby to the file name so it can be identified as a KookaBlockly file in the Kookaberry's menu

Additional Code



```
display text value= "C to send SOS"
    x= 0
    y= 10
    colour= 1
display text value= "A to Exit"
    x= 0
    y= 40
    colour= 1

when button A is pressed
    display clear
    display show
    exit program

every loop
    if button C is pressed
    do  turn pin P1 on
        sleep for 0.2 seconds
        turn pin P1 off
        sleep for 0.2 seconds
        turn pin P1 on
        sleep for 0.2 seconds
        turn pin P1 off
        sleep for 0.2 seconds
        turn pin P1 on
        sleep for 0.2 seconds
        turn pin P1 off
        sleep for 1 seconds
        turn pin P1 on
```

Kookaberry Screenshot



```
C to send SOS

A to Exit
```

Save it

Remember to save your app with the .kby suffix. Go to the KookaBlockly Support page for information on saving it in Kookaberry Scripts and on your Kookaberry.