IOT WORKSHEET 2 DOCUMENTATION

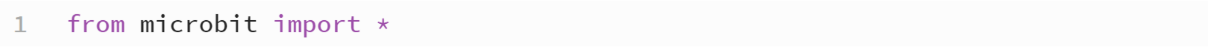
Shadhan Sidique

Morse code is used as the communication protocol to transmit data between 2 BBC Micro:bit devices using GPIOS and is implemented using Python.

Transmitter

The transmitter code is encrypted using Atbash cipher and then transmit it as Morse code signals through both visual and pin outputs.

Library Import:

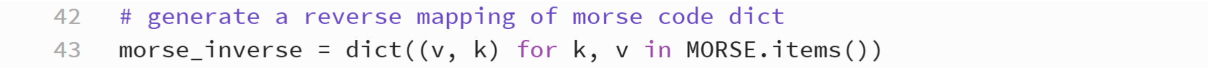


```python
from microbit import *
```

This imports everything from the microbit library thereby providing access to hardware functionalities of the BBC Micro:bit.

Morse code table contains Morse code mappings for both uppercase English alphabets and numbers from 0 to 9.

Reverse Morse Code Mapping:



```python
# generate a reverse mapping of morse code dict
morse_inverse = dict((v, k) for k, v in MORSE.items())
```

This line generates a reverse mapping where each English letter/number is mapped to its Morse code representation.

Configurable Constants:

These constants define how long the micro:bit displays and transmits each Morse code signal.

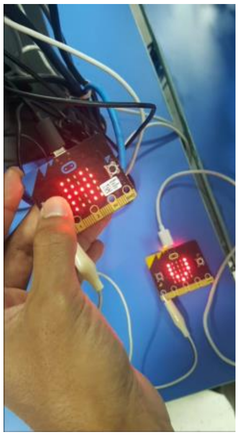
DOT_ACTIVATION defines the duration for dots (in milliseconds).

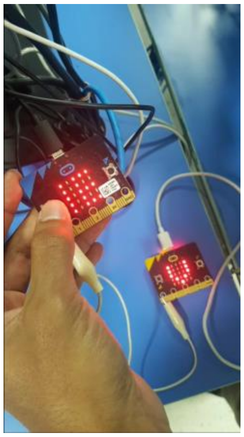
DASH_ACTIVATION defines the duration for dashes dots (in milliseconds).

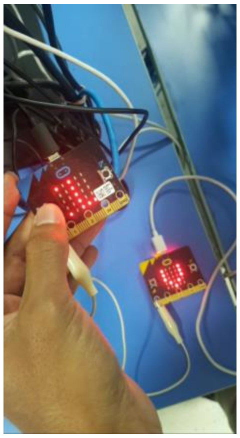
Atbash Cipher Function:

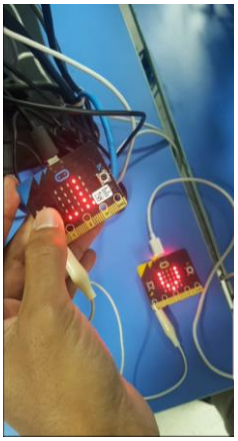
The **encrypt_message** function takes a message string and returns its encryption using Atbash cipher. In Atbash cipher, each alphabet is replaced by the letter in reverse position. For example: ‘A’↔’Z’, ‘B’↔’Y’.

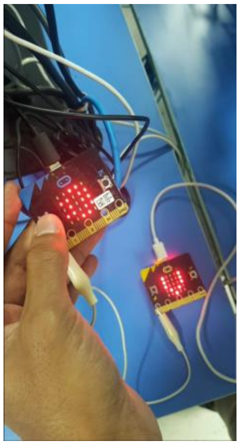
When button A on the Micro:bit is pressed, the code encrypts the word “HELLO” using Atbash cipher. The encrypted word is then transmitted as Morse code both visually (on the LED display of the Micro:bit) as well as electronically (using pin0).







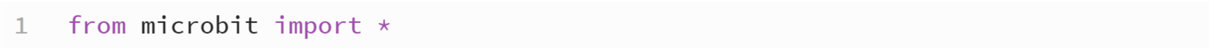




Receiver

The receiver code is also similar to the transmitter code but obviously it operates differently. It is set up to receive and interpret Morse code signals from pin0 and display the corresponding character on the Micro:bit.

Library Import:



```
from microbit import *
```

This imports everything from the microbit library thereby providing access to hardware functionalities of the BBC Micro:bit.

Morse code table contains Morse code mappings for both uppercase English alphabets and numbers from 0 to 9.

Configurable Constants:

These constants help to determine the nature of the signal received.

DOT_TIME is the expected maximum duration for a dot signal in milliseconds.

DASH_TIME is the expected maximum duration for a dash signal in milliseconds.

Initialization:

buffer is a string used to temporarily keep the Morse code symbols until a full letter or number is received.

wait_started is a timestamp of when waiting for a signal began.

Atbash Cipher Decryption Function:

The decrypt_message function decrypts the encrypted message using Atbash cipher. In Atbash cipher, each alphabet is replaced by the letter in reverse position. For example: ‘A’↔’Z’, ‘B’↔’Y’.

Buffer and Display:

The received Morse code symbol is appended to the **buffer**. If there is a long pause between the signals (greater than **DASH_TIME**) or if the **buffer** contains 5 characters then the code interprets the content of the **buffer** as a Morse-coded character.

The Morse-coded character is looked up in the MORSE dictionary to find its equivalent English uppercase character or number.

- This character is then decrypted using Atbash cipher.
- The decrypted character is displayed on the LED display of the BBC Micro:bit device.
- The **buffer** is cleared to be ready for the next set of signals.

Overall, this code sets up the BBC Micro:bit to act as a Morse code receiver. It listens for Morse code signals on pin0, interprets them, decrypts them using Atbash cipher and displays the resulting characters on the LED display of the BBC Micro:bit device.