

**Worksheet 1 - Basic micro:GUI coded in 'Block'**

by Innovations in Education

3rd February 2018

**micro:GUI Worksheet Overview**

The aim of this Worksheet is to help you understand the **micro:GUI specification** and learn how the **micro:GUI** works. You are then challenged to write the code for your own '***Basic***' **micro:GUI** in '**Block'.**

By following this Worksheet you (or your students) will start to understand the concepts of a **Graphical User Interface** **(GUI)**. You will see how a **GUI** makes it easier for users to interact with a computer and appreciate what features make a **GUI** ***intuitive to use***.

The Worksheet assumes that you (or your students) are already familiar with programming the **micro:bit** in '**Block**'

**What you will need:-**

* A **PC, Mac** or **Tablet** set up to program in '**Block**' using the '**MakeCode**' (previously known as PXT) Editor (<https://makecode.microbit.org/>)
* A **micro:bit** with USB cable
* A **speaker** attached to **micro:bit** GPIO **P0** (*Optional*)
* A PDF copy of the **micro:GUI specification** downloaded from

<http://www.zbit-connect.co.uk/microGUI/>

* A copy of the **example program** ***'microbit-microGUI-Block-Example.hex'*** downloaded from <http://www.zbit-connect.co.uk/microGUI/>

**Introduction Step 1**:- The best way to understand the **micro:GUI specification** is to start by loading the '***microGUI-Block-Example.hex'*** file onto your **micro:bit** and try it out!

When the program starts the **Cursor** **LED** **[C]** should flash in the top left corner of the **micro:bit's** display and ***3* micro:App** **LEDs** should be illuminated in the bottom left.

Use **micro:bit** **Buttons <A>** or **<B>** to move the **Cursor**. See how you can move the **Cursor** to anywhere on the display using a maximum of 8 Button presses.

Move the **Cursor** onto one of the **micro:App** **LED's** and see how the App's '**micro:Icon'** is displayed.

When the **Cursor** is on a **micro:App LED**, press **Buttons <A+B>** together to 'launch' the **micro:App**. Each **micro:App** in the ***'micro:GUI-PXT-example.hex'*** file plays a different tune and scrolls the **micro:App** number across the screen.

To return to the '**Home Screen'** at any time, press the **<Reset>** Button

**Introduction Step 2**:- *Now* read the **micro:GUI specification**.

Now that you've tried using the **micro:GUI**, the **specification** should make more sense!

Note that when coding in **Block**, **JavaScript** or **Python** the LED's on the **micro:Bit's** display are numbered using (**x, y**) co-ordinates as shown with:-

* (0, 0) the top left LED
* (4, 0) the top right LED
* (0, 4) the bottom left LED
* (4, 4) the bottom right LED,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **x** | **0** | **1** | **2** | **3** | **4** | **y** |  |
|  |  |  |  |  |  |  | **0** |  |
|  |  |  |  |  |  |  | **1** |  |
| **A** |  |  |  |  |  |  | **2** | **B** |
|  |  |  |  |  |  |  | **3** |  |
|  |  |  |  |  |  |  | **4** |  |

**Coding Step 1**:- Write the code to **control the Cursor**

*Try and write the code yourself but if you get stuck there is some 'Step 1' example code at the end of the Worksheet.*

The code needs to:-

* Define a pair of variable for the **x** and **y** the co-ordinates of the **Cursor** and initialise them to (0,0)
* Define a **function** that updates the **Display** by using **'clear screen'** then **'Plot x y'** to illuminate the Cursor LED
* Call this **function** in a **'forever'** loop
* Add one to the variable for the **x** co-ordinate on Button **<A>** Press
* Add one to the variable for the **y** co-ordinate on Button **<B>** Press
* When the **x** co-ordinate reaches 5 the Cursor will go off the right edge of the screen so set it back to 0 so it reappears on the left
* Similarly when the **y** co-ordinate reaches 5 the Cursor will go off the bottom edge of the screen so set it back to 0 so it reappears at the top

Do you think it is more intuitive to have the **<A>** or **<B>** Button moving the cursor '***Right***'? If you think it is more intuitive to use the **<B>** Button to move the cursor '***Right***' then change it!

* If you have a speaker attached to your **micro:bit**, add **'play tone'** whenever the **Cursor** moves, using a different tone when it moves **x** and **y** direction

**Coding Step 2**:- Write the code to **display the micro:App LED's**

You will need to:-

* Modify the **Display** **function** by using **'Plot x y'** to illuminate each of the **micro:App LED's** along the bottom of the display.

**Coding Step 3**:- Write the code to **display the micro:Icons**

You will need to:-

* When the **Button** **<A>** or **<B>** is presses and the **Cursor** co-ordinates are changed, add a check to see if the new co-ordinates are on a **micro:App LED**. If so set a **variable** called **'icon\_on'** to **'true'**. If not set this **variable** to **'false'.** (This variable is known as a **'Flag'**)
* In the **'forever'** loop, if the **'icon\_on'** flag is set, call a ***new* function** called **'icon'** to display the **icon**, otherwise call the function to display the normal **micro:GUI** screen
* Add the new **'icon' function** using **'if, else if'** statements to check which **micro:App LED's** the **Cursor** is on and if display a suitable **icon** (Block provides a number of icons in the '**Basic**' menu.

**Coding Step 4**:- Write the code to **launch the micro:Apps**

You will need to:-

* Add code for when **Buttons** **<A+B>** are pressed. This code should check the co-ordinates of the **Cursor** and call the associated **micro:App function.** e.g. **function** **'App0'** for **micro:App 0, function 'App1'** for **micro:App 1,** etc

**Coding Step 5**:- Write the **code for the micro:Apps**

Finally you will need to:-

* Write the code for each of your **micro:Apps** in **functions** **'App0', 'App1',** etc!

(A simple **micro:App** might be to play a tune (melody) from the '**music**' menu and scroll a message across the display while the tune plays)

*You now have a working* ***micro:GUI***!

***...but can you make it even better?***

**Example Code for Step 1 - Controlling the Cursor**



**Example Code for Step 5 - Fully working *basic* micro:GUI in 'Block'**

The code for ***'microbit-microGUI-Block-Example'*** can be found on:-

<http://www.zbit-connect.co.uk/microGUI/>

Note that the example code includes code to **'blink'** the **Cursor** which make the **Cursor** stand out.

How could you modify your code to ***make the Cursor blink***?