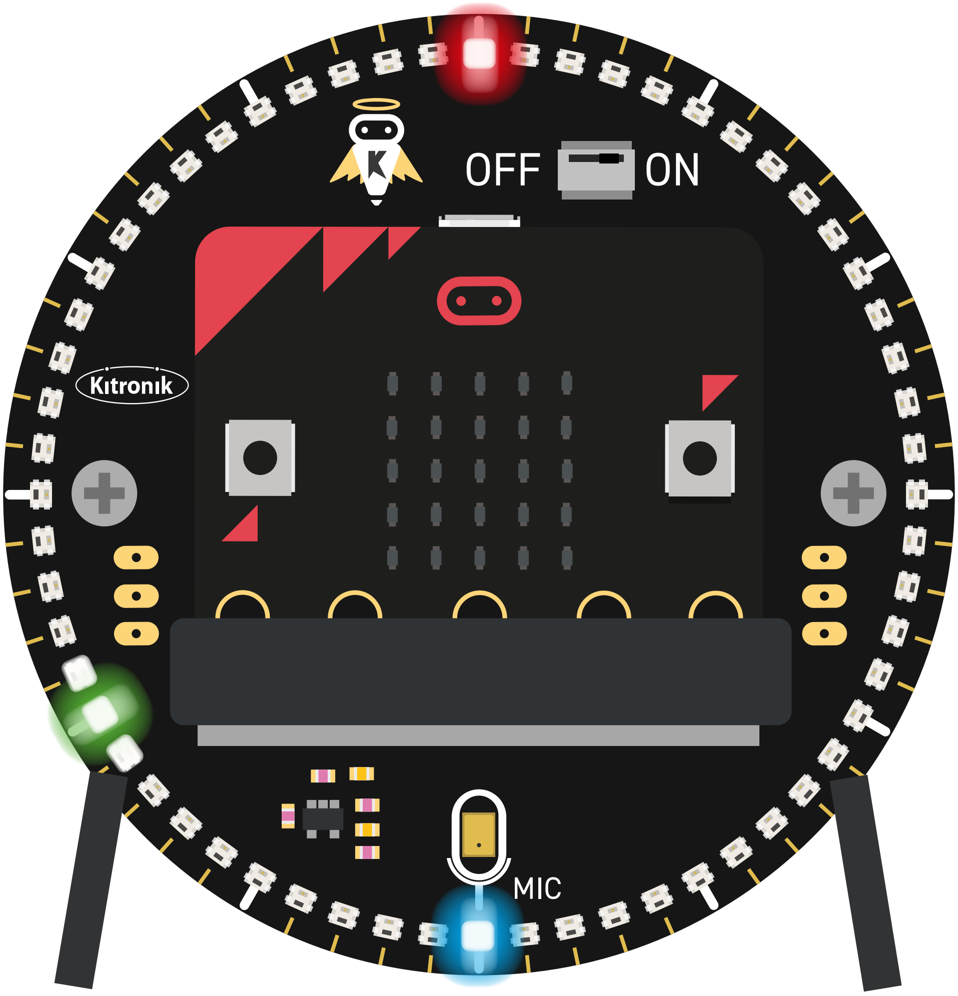
HALO HD WAKE UP LIGHT ALARM CLOCK TUTORIAL

KITRONIK **RESOURCES**



INTRODUCTION

Learn how to create a bedside clock which gently fills the room with light as wake up time approaches.
If you have not done the 'Halo HD Basic Clock', 'Halo HD Adjustable Clock' & 'Halo HD Alarm Clock' tutorials, it is recommended that you complete them first.

SETTING UP

EQUIPMENT REQUIRED:

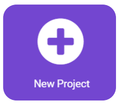
• 1 x BBC micro:bit (www.kitronik.co.uk/5613),
• 1 x Halo HD - Alarm Clock Kit (www.kitronik.co.uk/5681)

ADDING IN CUSTOM MAKECODE BLOCKS:

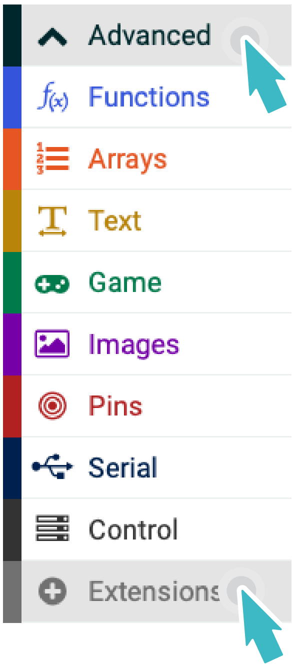
We have made custom coding blocks especially for the Halo HD, which helps to make coding super simple within Microsoft MakeCode.
To add these blocks, follow the steps below:

STEP 1: Bring up the MakeCode Block Editor - (makecode.microbit.org).

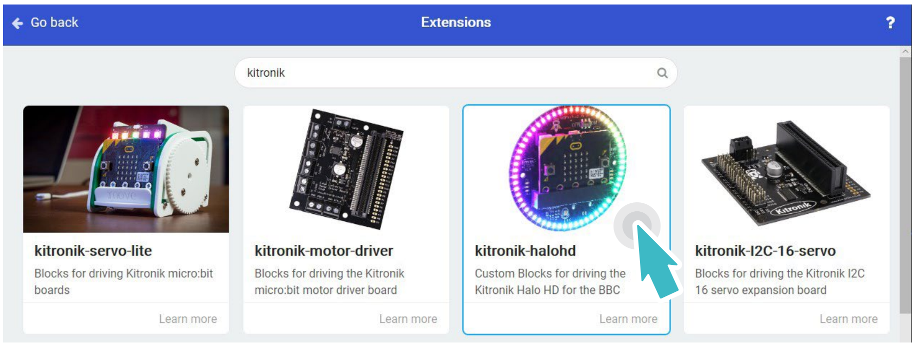
STEP 2: Click 'New Project'.



STEP 3: In the toolbox towards the left of the screen, select the '**Advanced**' section. Additional block categories will appear below.



STEP 4: Select '**Extensions**'.
STEP 5: In the pop up's search bar type '**Kitronik**'.
STEP 6: Locate & select the '**kitronik-halohd**' box.


Go back
Extensions
?
kitronik
kitronik-servo-lite
Blocks for driving Kitronik micro:bit boards
Learn more
kitronik-motor-driver
Blocks for driving the Kitronik micro:bit motor driver board
Learn more
kitronik-halohd
Custom Blocks for driving the Kitronik Halo HD for the BBC
Learn more
kitronik-I2C-16-servo
Blocks for driving the Kitronik I2C 16 servo expansion board
Learn more

THE TUTORIAL

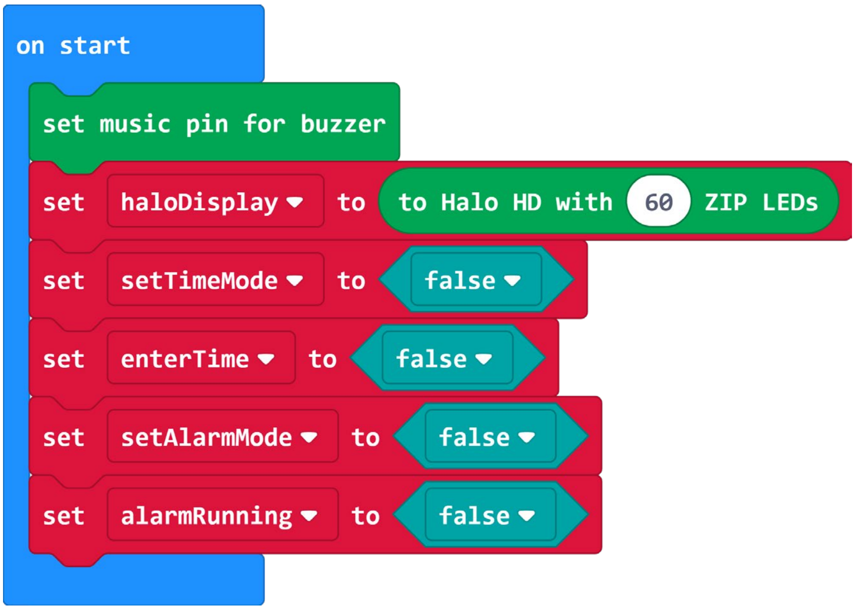
RECREATE ALARM CLOCK CODE

STEP 1: This tutorial carries on from the 'Halo HD Alarm Clock' tutorial, so start by opening your program from the 'Adjustable Clock Tutorial' and refamiliarising yourself with it.

INCREASING BRIGHTNESS

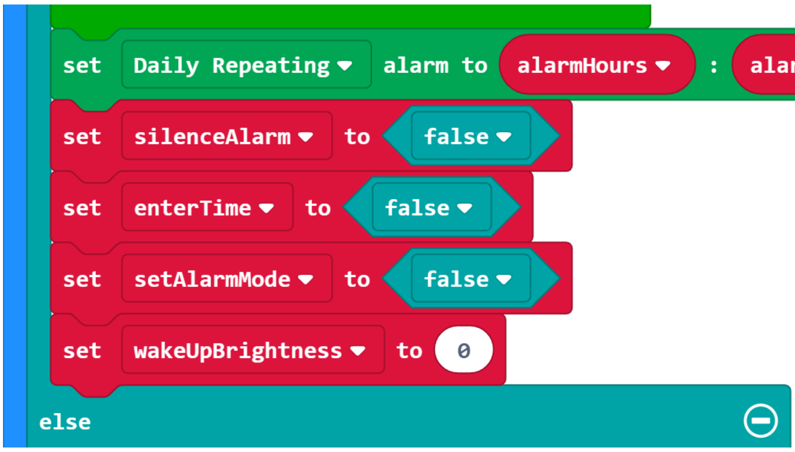
STEP 1: Rather than the simple tune playing as an alarm, this time we want to use the lights on the Halo HD as a Wake Up Light alarm. Wake Up Lights are a simulation of the sun rising and light getting brighter outside, which makes waking up more natural.

To get started, make two new variables: 'alarmRunning' and 'wakeUpBrightness'. Set 'alarmRunning' to be 'false' in the 'on start' block.



```
on start
    set music pin for buzzer
    set haloDisplay to to Halo HD with 60 ZIP LEDs
    set setTimeMode to false
    set enterTime to false
    set setAlarmMode to false
    set alarmRunning to false
```

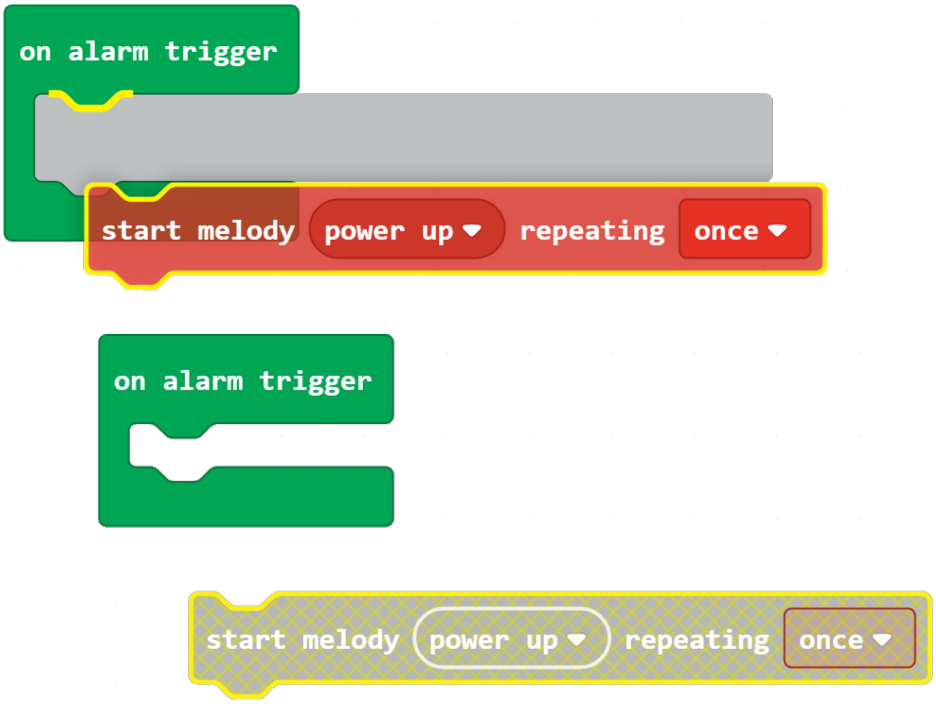
STEP 2: In the 'forever' loop, change the 'set alarm' block so that it is now a "Daily Repeating" alarm which is "Auto" silenced, and set 'wakeUpBrightness' to be 0.



```
set Daily Repeating ▾ alarm to alarmHours ▾ : alarmMinutes ▾ with Auto ▾ Silence
set silenceAlarm ▾ to false ▾
set enterTime ▾ to false ▾
set setAlarmMode ▾ to false ▾
set wakeUpBrightness ▾ to 0
else
```

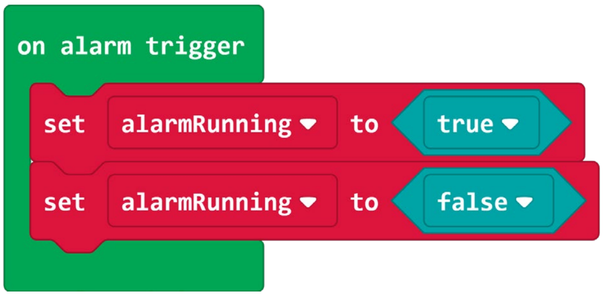
STEP 3: Next, remove the extra 'if' statement at the end of the 'forever' loop checking whether 'silenceAlarm' is 'true'.

STEP 4: Now we need to change what happens when the alarm triggers. To make things easier, temporarily remove the 'start melody' block from inside the 'on alarm trigger' block.



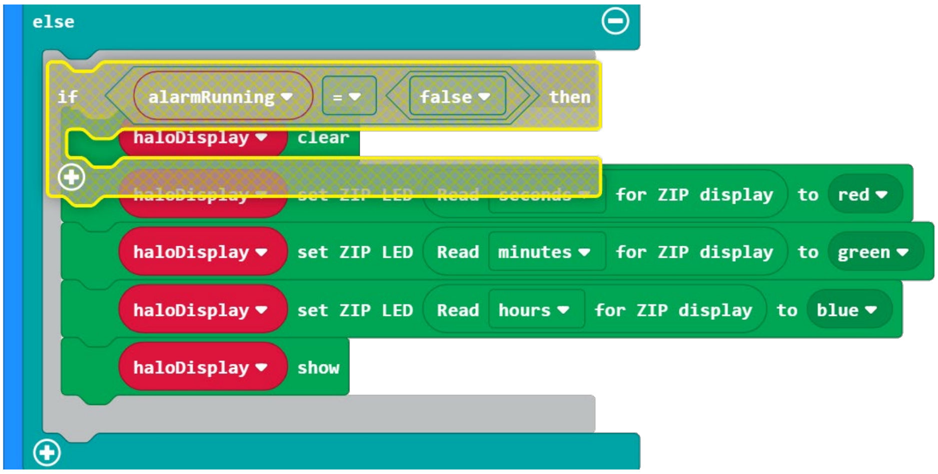
```
on alarm trigger
    start melody power up ▾ repeating once ▾

on alarm trigger

start melody power up ▾ repeating once ▾
```

STEP 5: This is where we use our 'alarmRunning' variable. Set it to be 'true' at the start of the 'on alarm trigger' block, and 'false' at the end (the rest of our code will go inbetween). This variable will be able to tell us at any point whether the alarm is triggered or not. The next step will show why this is important...

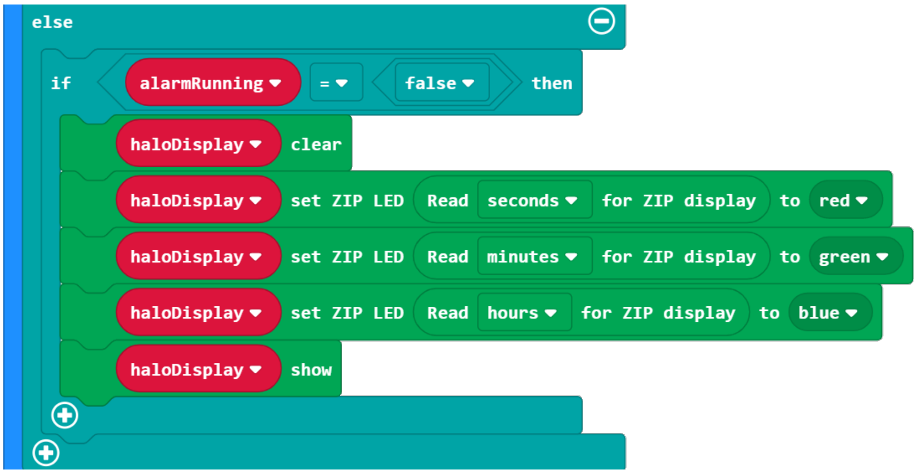


```
on alarm trigger
    set alarmRunning to true
    set alarmRunning to false
```

STEP 6: Our Wake Up Light alarm will make use of all the Halo HD LEDs, which means the time will not be able to display while they are in use. To make sure there is no clash in the code, we need to check whether the alarm is triggered at some point in the 'forever' loop. In the 'else' section, put all the code inside it in an 'if' statement checking whether 'alarmRunning' is 'false'. This way, the time display code will only run when the alarm is not triggered.

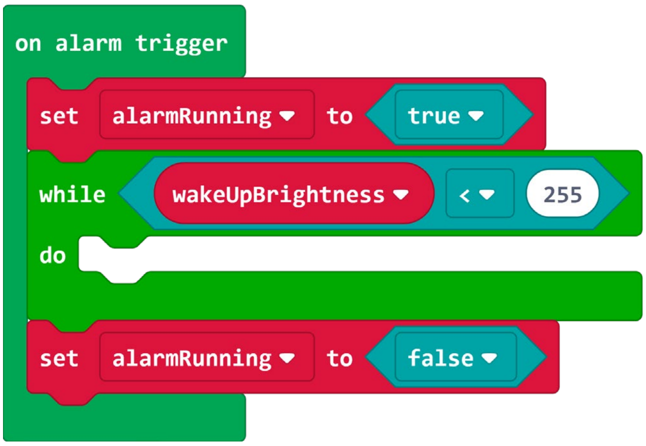


```
else
    if alarmRunning = false then
    haloDisplay clear
    haloDisplay set ZIP LED Read seconds for ZIP display to red
    haloDisplay set ZIP LED Read minutes for ZIP display to green
    haloDisplay set ZIP LED Read hours for ZIP display to blue
    haloDisplay show
```



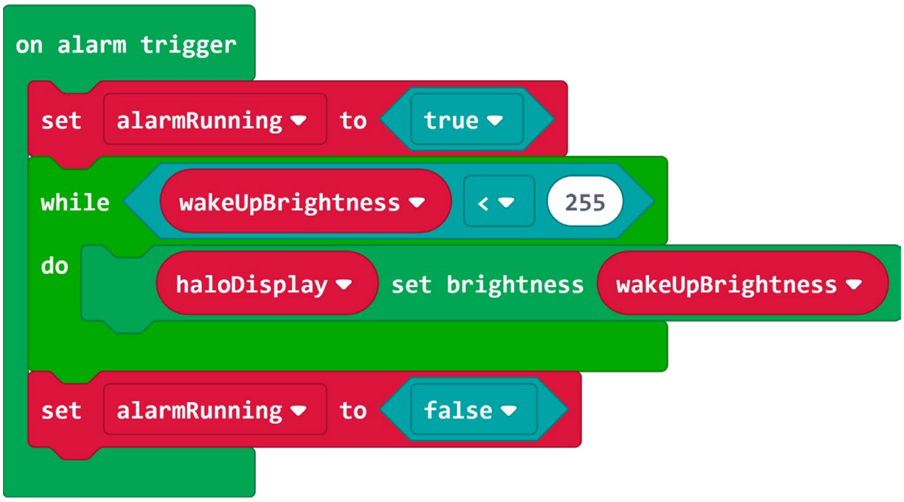
```
else
    if alarmRunning = false then
        haloDisplay clear
        haloDisplay set ZIP LED Read seconds for ZIP display to red
        haloDisplay set ZIP LED Read minutes for ZIP display to green
        haloDisplay set ZIP LED Read hours for ZIP display to blue
        haloDisplay show
```

STEP 7: Back to the alarm triggering; time to add in the light control. For a Wake Up Light, the LEDs need to start off and gradually increase in brightness. To do this, we will use another 'while' loop, which should continue to run while 'wakeUpBrightness' is less than 255.



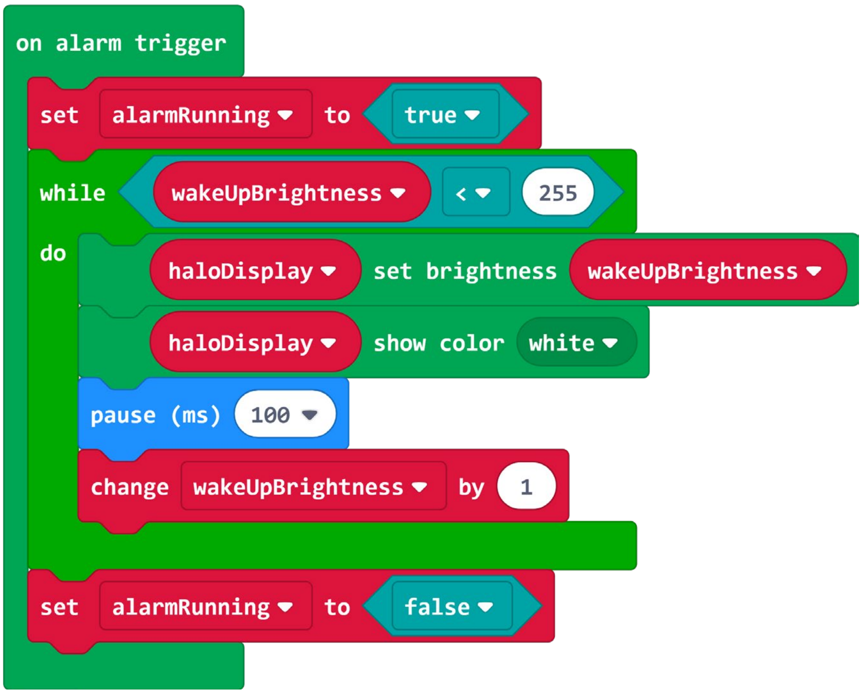
```
on alarm trigger
    set alarmRunning to true
    while wakeUpBrightness < 255
    do
    set alarmRunning to false
```

STEP 8: Inside our new 'while' loop, start by adding the 'set brightness' block from the ZIP LEDs section of the Halo HD extension, and then set the brightness to the variable 'wakeUpBrightness'.



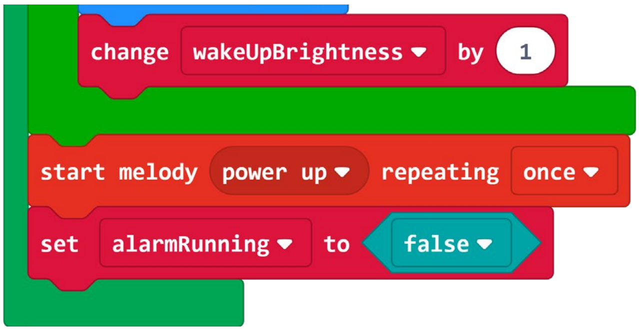
```
on alarm trigger
    set alarmRunning to true
    while wakeUpBrightness < 255
    do
        haloDisplay set brightness wakeUpBrightness
    set alarmRunning to false
```

STEP 9: Next, use the 'show color' block from the ZIP LEDs section of the Halo HD extension to display all the LEDs white, and then add a 100ms 'pause' followed by a 'change wakeUpBrightness by 1' block.



```
on alarm trigger
    set alarmRunning to true
    while wakeUpBrightness < 255
    do
        haloDisplay set brightness wakeUpBrightness
        haloDisplay show color white
        pause (ms) 100
        change wakeUpBrightness by 1
    set alarmRunning to false
```

STEP 10: Finally, remember that 'start melody' block we removed earlier? Drag that back into the 'on alarm trigger' block just after the 'while' loop.



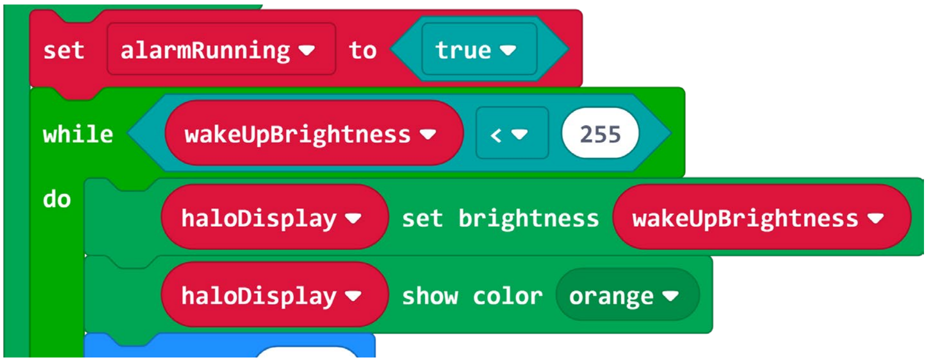
```
        change wakeUpBrightness by 1
    start melody power up repeating once
    set alarmRunning to false
```

STEP 11: If you have a micro:bit connected, click 'Download' to transfer your code, set another alarm and check the brightening lights work.

CODING A SUNRISE

We now have an alarm that gradually gets brighter when it triggers, but the light is very harsh, and doesn't really replicate dawn very well. What we need to do is to code a sunrise.

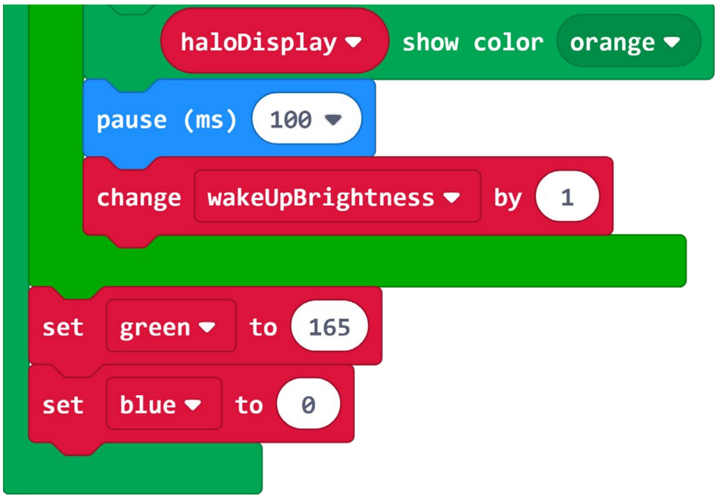
STEP 1: If you look at a sunrise, you'll see that it starts with much more red light, and as it gets brighter, more blue light comes in. To simulate this, we're going to start by displaying orange, progress through to yellow, and finish with white.
To get started, change the 'show color' block in the 'on alarm trigger' section to display "orange" rather than "white".



```
set alarmRunning to true
while wakeUpBrightness < 255
do
    haloDisplay set brightness wakeUpBrightness
    haloDisplay show color orange
```

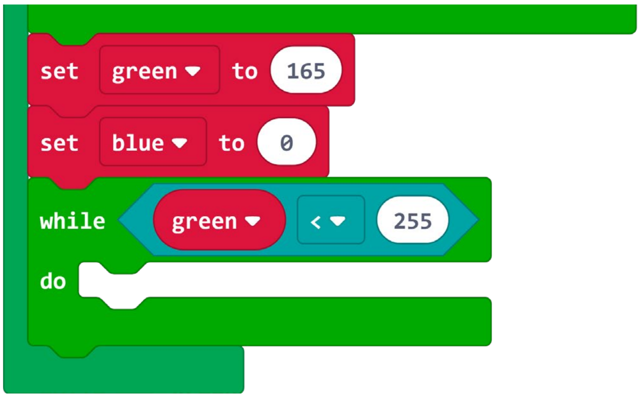
STEP 2: Then we need to temporarily remove the 'start melody' and 'set alarmRunning to false' blocks, just like before.

STEP 3: Now we need to create two new variables, 'green' and 'blue'. These will be used for the ZIP LED colour settings. Just after the 'while' loop in the 'on alarm trigger' block, set 'green' to be 165 and 'blue' to be 0. (With red at 255, this RGB colour combination will form orange).



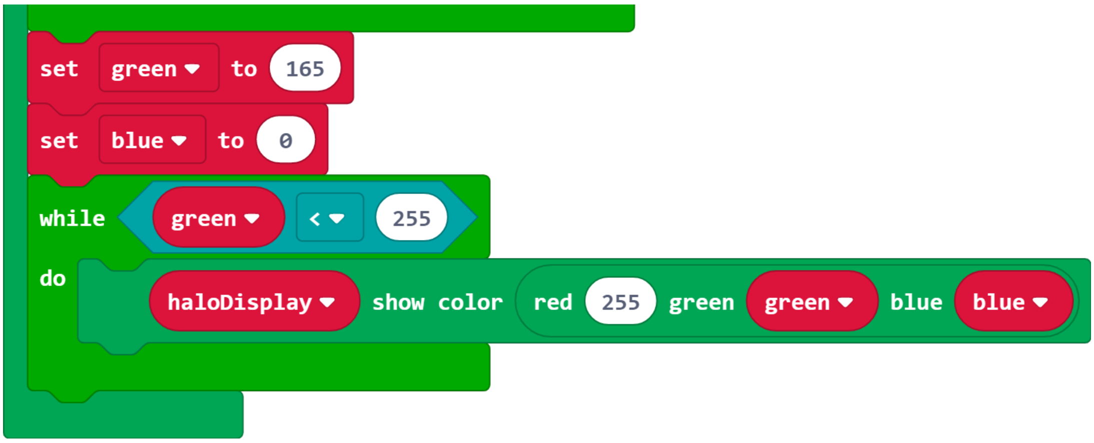
```
    haloDisplay show color orange
    pause (ms) 100
    change wakeUpBrightness by 1
set green to 165
set blue to 0
```

STEP 4: Next, add a second 'while' loop, one that continues while 'green' is less than 255.



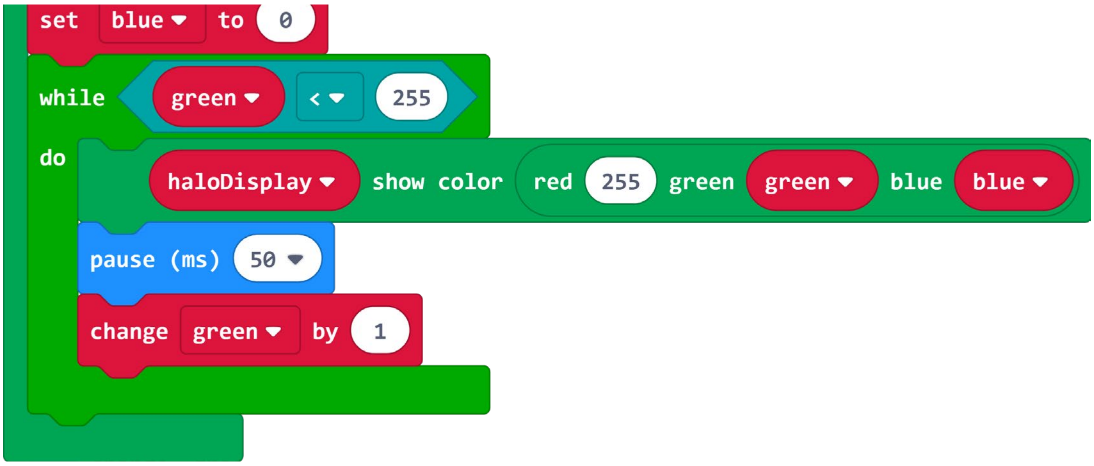
```
set green ▾ to 165
set blue ▾ to 0
while green ▾ < ▾ 255
do
```

STEP 5: Inside the new 'while' loop, add a 'show color' block displaying a colour set by the RGB values block. Set the "red" value to 255, the "green" value to the variable 'green' and the 'blue' value to the variable 'blue'.



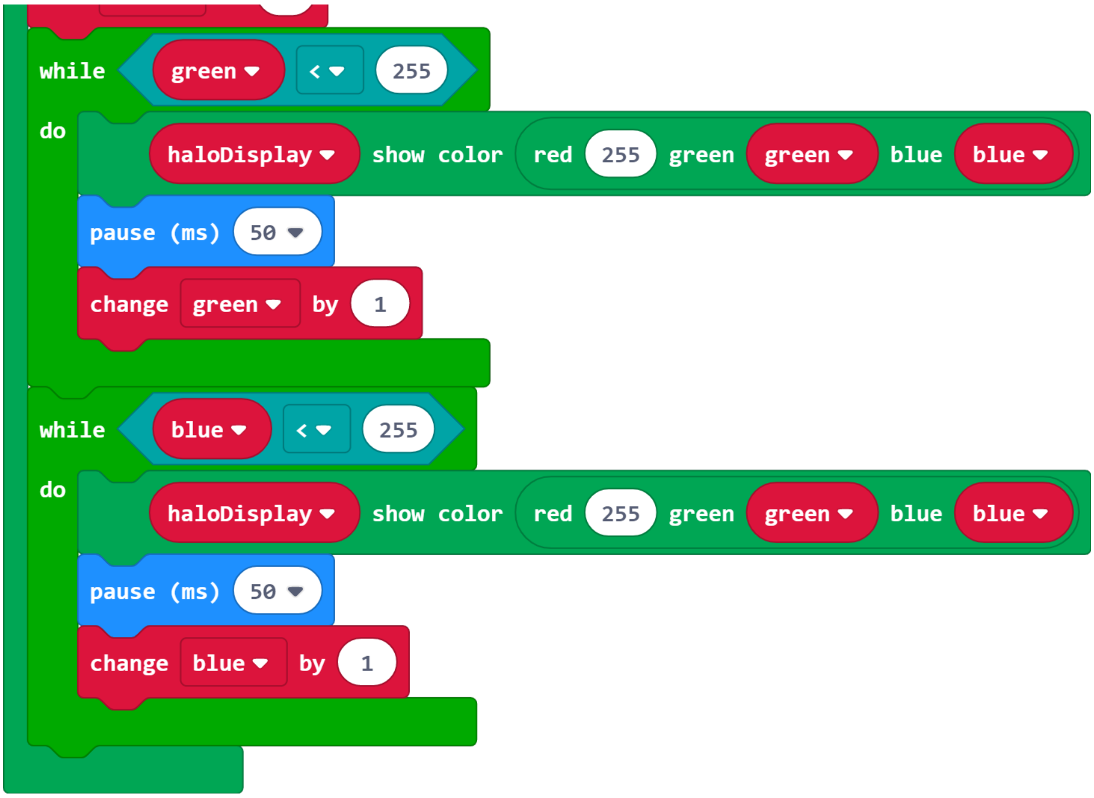
```
set green ▾ to 165
set blue ▾ to 0
while green ▾ < ▾ 255
do
    haloDisplay ▾ show color red 255 green green ▾ blue blue ▾
```

STEP 6: The final things to add to 'while' loop is a 50ms 'pause' and a block to 'change green by 1'.



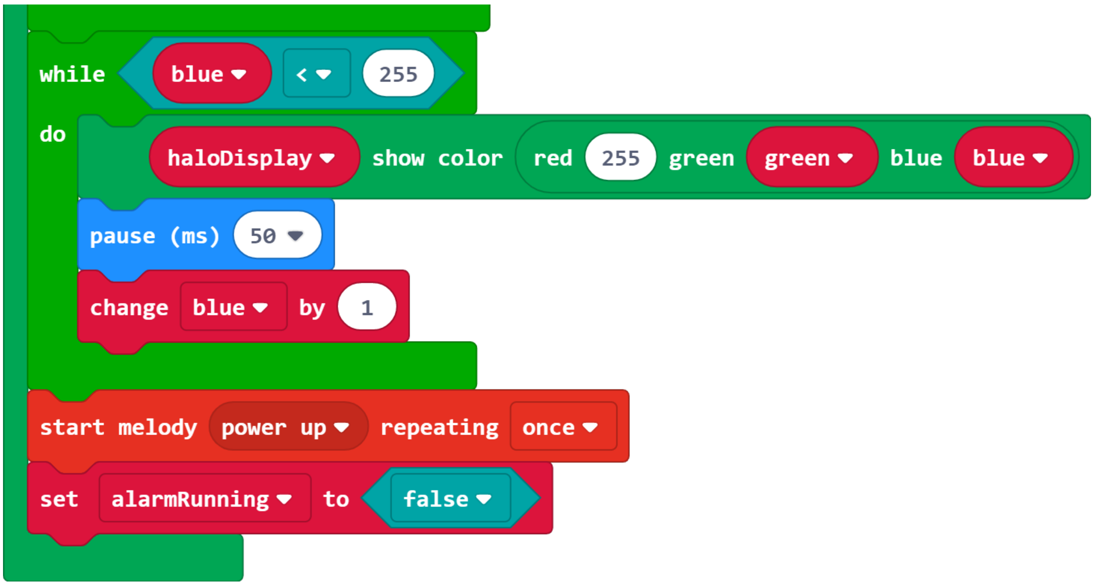
```
set blue ▾ to 0
while green ▾ < ▾ 255
do
    haloDisplay ▾ show color red 255 green green ▾ blue blue ▾
    pause (ms) 50 ▾
    change green ▾ by 1
```

STEP 7: Repeat the previous three steps, but this time the 'while' loop should continue while the value of 'blue' is less than 255, and it should also be 'blue' which should change by 1 at the end of the loop.



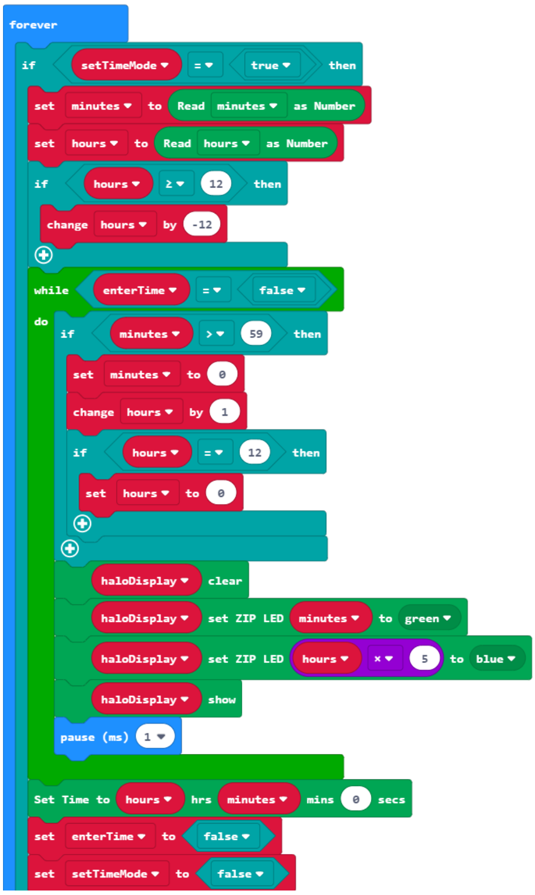
```
while green < 255
do
    haloDisplay show color red 255 green green blue blue
    pause (ms) 50
    change green by 1
while blue < 255
do
    haloDisplay show color red 255 green green blue blue
    pause (ms) 50
    change blue by 1
```

STEP 8: Finally, drag the 'start melody' and 'set alarmRunning to false' blocks we removed earlier back into the 'on alarm trigger' block just after the third 'while' loop.

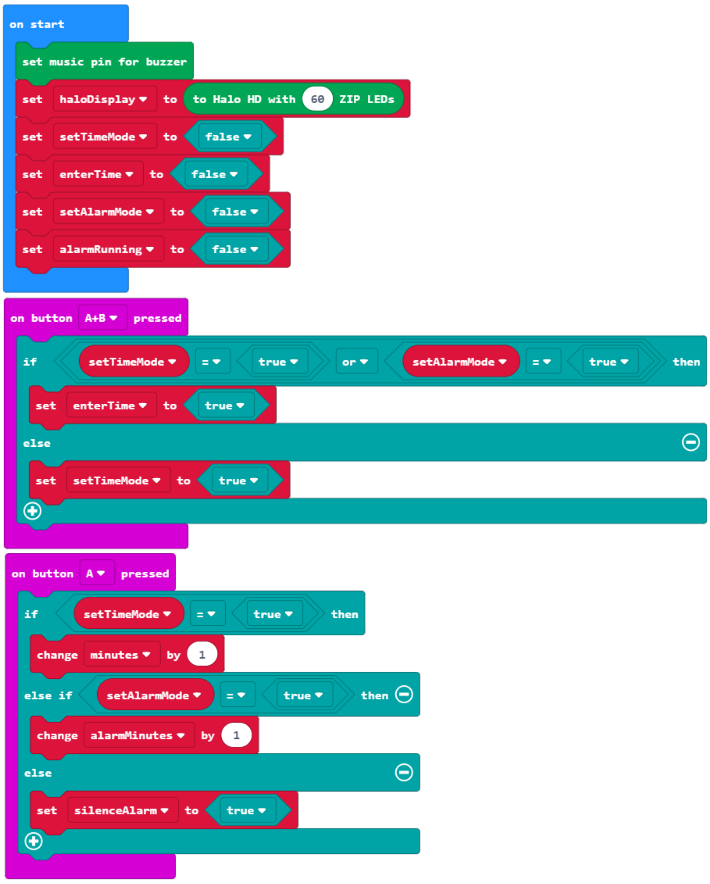


```
while blue < 255
do
    haloDisplay show color red 255 green green blue blue
    pause (ms) 50
    change blue by 1
start melody power up repeating once
set alarmRunning to false
```

STEP 9: We've done it! A full sunrise simulation alarm clock. 'Download' your code and transfer it to your Halo HD and try it out. (The complete code is shown across the next two pages).

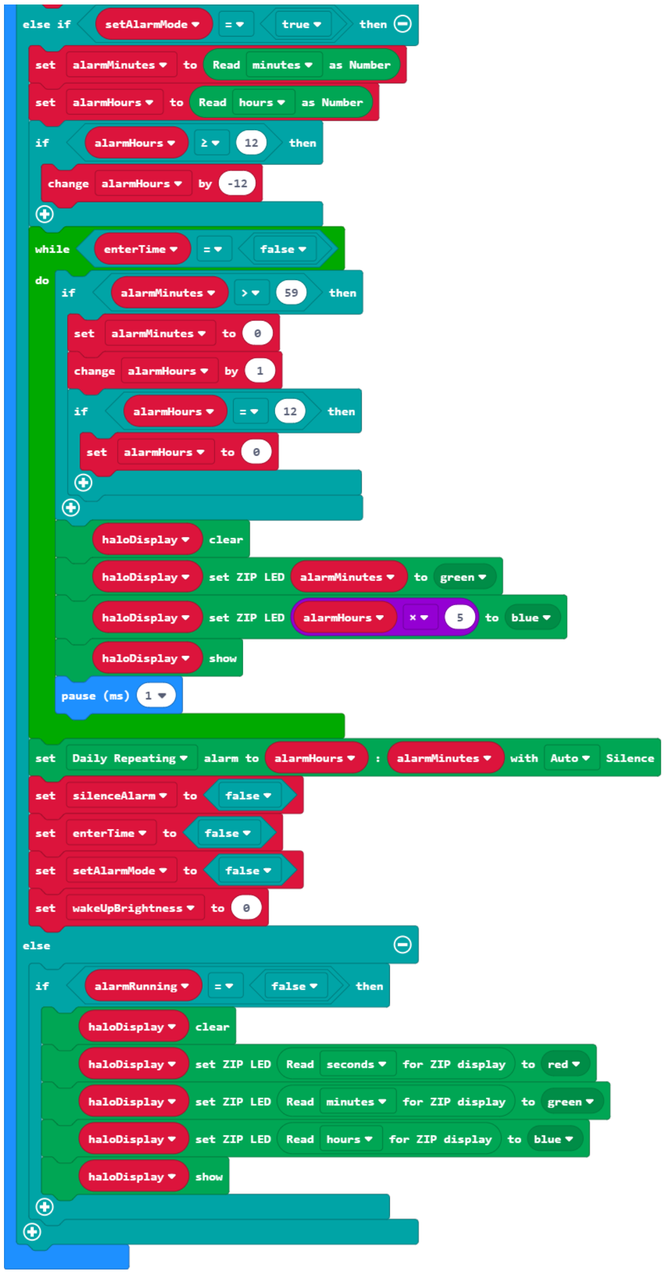


```
forever
    if setTimeMode = true then
        set minutes to Read minutes as Number
        set hours to Read hours as Number
        if hours ≥ 12 then
            change hours by -12
    while enterTime = false
    do
        if minutes > 59 then
            set minutes to 0
            change hours by 1
            if hours = 12 then
                set hours to 0
        haloDisplay clear
        haloDisplay set ZIP LED minutes to green
        haloDisplay set ZIP LED hours × 5 to blue
        haloDisplay show
        pause (ms) 1
    Set Time to hours hrs minutes mins 0 secs
    set enterTime to false
    set setTimeMode to false
```

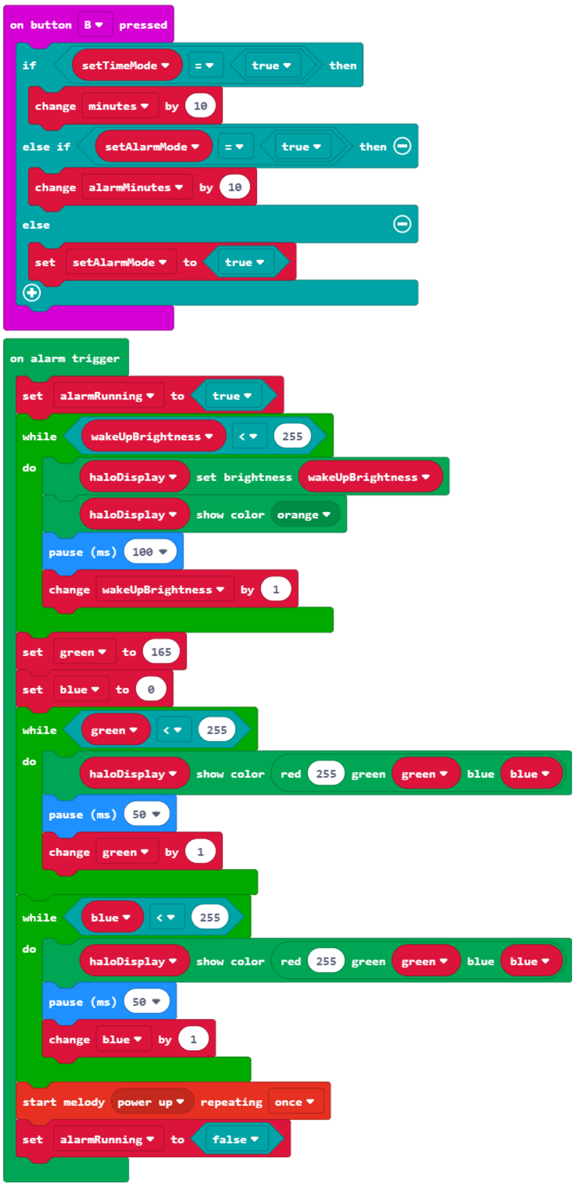


```
on start
    set music pin for buzzer
    set haloDisplay to to Halo HD with 60 ZIP LEDs
    set setTimeMode to false
    set enterTime to false
    set setAlarmMode to false
    set alarmRunning to false

on button A+B pressed
    if setTimeMode = true or setAlarmMode = true then
        set enterTime to true
    else
        set setTimeMode to true

on button A pressed
    if setTimeMode = true then
        change minutes by 1
    else if setAlarmMode = true then
        change alarmMinutes by 1
    else
        set silenceAlarm to true
```

COMPLETE CODE CONTINUED...



```
else if setAlarmMode = true then
    set alarmMinutes to Read minutes as Number
    set alarmHours to Read hours as Number
    if alarmHours ≥ 12 then
        change alarmHours by -12
    while enterTime = false
    do  if alarmMinutes > 59 then
            set alarmMinutes to 0
            change alarmHours by 1
            if alarmHours = 12 then
                set alarmHours to 0
        haloDisplay clear
        haloDisplay set ZIP LED alarmMinutes to green
        haloDisplay set ZIP LED alarmHours × 5 to blue
        haloDisplay show
        pause (ms) 1
    set Daily Repeating alarm to alarmHours : alarmMinutes with Auto Silence
    set silenceAlarm to false
    set enterTime to false
    set setAlarmMode to false
    set wakeUpBrightness to 0
else
    if alarmRunning = false then
        haloDisplay clear
        haloDisplay set ZIP LED Read seconds for ZIP display to red
        haloDisplay set ZIP LED Read minutes for ZIP display to green
        haloDisplay set ZIP LED Read hours for ZIP display to blue
        haloDisplay show
```



```
on button B pressed
    if setTimeMode = true then
        change minutes by 10
    else if setAlarmMode = true then
        change alarmMinutes by 10
    else
        set setAlarmMode to true

on alarm trigger
    set alarmRunning to true
    while wakeUpBrightness < 255
    do  haloDisplay set brightness wakeUpBrightness
        haloDisplay show color orange
        pause (ms) 100
        change wakeUpBrightness by 1
    set green to 165
    set blue to 0
    while green < 255
    do  haloDisplay show color red 255 green green blue blue
        pause (ms) 50
        change green by 1
    while blue < 255
    do  haloDisplay show color red 255 green green blue blue
        pause (ms) 50
        change blue by 1
    start melody power up repeating once
    set alarmRunning to false
```

For any further queries or support, please visit the Kitronik website: www.kitronik.co.uk/5672

Or get in touch:

Telephone	+44 (0) 115 970 4243
Sales email:	sales@kitronik.co.uk
Tech Support email:	support@kitronik.co.uk
Web:	www.kitronik.co.uk

kitronik.co.uk/twitter
kitronik.co.uk/facebook
kitronik.co.uk/youtube
kitronik.co.uk/instagram

Designed & manufactured
in the UK by Kitronik

RoHS