Read Temperature / Select Case

main:

readtemp c.0, b1 ' read a temperature input from pin C.0

'and store value in degrees C as variable b

Select case b1

case >**10**

high B.1 'turn on output B1 if the temperature is higher than 10 degrees

case >**15**

high B.2 'turn on output B2 if the temperature is higher than 15 degrees

Case >**20**

high B.3 'turn on output B3 if the temperature is higher than 20 degrees

case >**25**

high B.4 'turn on output B4 if the temperature is higher than 25 degrees

End select

pause **1000**

goto main

#no\_data 'reduce download time

**Display Text on OLED**

init: pause **500** ; wait for display to initialise

main: serout B.7,N2400,(**254**,**128**) ; move to start of first line

serout B.7,N2400,(**254**,**1**) ; Clear display

pause **500**

serout B.7,N2400,("Hello") ; output text

pause **500**

serout B.7,N2400,(**254**,**192**) ; move to start of second line

serout B.7,N2400,("Hello") ; output text

pause **2000**

goto main:

end

Display minutes and seconds on OLED

init:

pause **500** ; wait for display to initialise

main:

'serout B.7,N2400,(254,128) ; move to start of first line

serout B.7,N2400,(**254**,**1**) ; Clear display

pause **1000**

for minute = **0** to **5**

serout B.7,N2400,(**254**,**1**) ; Clear display

pause **1000**

For b1 = **1** to **60**

pause **50**

serout B.7,N2400,(**254**,**128**) ; move to start of first line

pause **50**

bintoascii minute,b6,b7,b8 'turn the binary number of minutes into decimal ascii characters

serout B.7,N2400,(b8," minutes") ; output minutes elapsed

bintoascii b1,b2,b3,b4 'turn the binary number of seconds into decimal ascii characters

serout B.7,N2400,(**254**,**192**) ; move to start of second line

serout B.7,N2400,(b3,b4," seconds") ; output seconds elapsed

pause **900**

Next b1

next minute

goto main:

end

Use a button input to set a variable representing temperature

main:

settemp:

if pinC.1 = **1** then pause **100** 'use pin C.1 input to set variable varA to 15

let varA = **15**

pause **100**

endif

if pinC.7 = **1** then pause **100** 'use pin C.7 input to set variable varA to 25

let varA = **25**

pause **100**

endif

readtemp c.0, b1 ' read a temperature input from pin C.0

'and store value in degrees C as variable b

Select Case b1

case >varA

high B.0 'turn on LED to indicate high temperature reading

case <varA

Low B.0 'make sure B.0 LED is turned off

High B.1 'turn on LED to indicate low temperature reading

End select

pause **1000**

goto main

#no\_data 'reduce download time

Use potentiometer to input a number / Select Case

'This code should read the voltage at pin A2 which will be somewhere between 0 and 5 volt. The chip reads this as a number between 0 and 256 according to the voltage. If you connect the potentiometer input 'pot' to C3 and adjust the potentiometer (blue component next to LEDS and switches) you can set the voltage at pin C3. The chip will read 0 to 1 volts as 0 to 50, 1 to 2 volts as 50 to 100 etc..

'depending on the potentiometer setting, the select case statement sends the program to minute 2,3,4,5 or 6 and then executes the code there (i.e turns on an output pin) before going on to startcooking.

main: ; start program

readadc A.2, varA

select case varA

case <**50**

pause **100**

goto minute2:

case <**100**

pause **100**

goto minute3:

case <**150**

pause **100**

goto minute4:

case <**200**

pause **100**

goto minute5:

case <**257**

pause **100**

goto minute6:

End select

goto main

minute2:

high B.2

goto startcooking

minute3:

high B.3

goto startcooking

Control Servo

servpos = **150** ' set servo to central position

servo B.0, servpos

for cntB = **150** to **220** step **5** ' loop to move servo 5 degrees every i/2 second

servpos = cntB

servo B.0, servpos

pause **500**

next

servpos = **150** ' put servo back to central osition

for cntB = **150** to **80** step **-5** ' loop to move servo 5 degrees every i/2 second

servpos = cntB

servo B.0, servpos

pause **500**

next

servpos = **150** ' put servo back to central osition

Use button input to make LED flash after button is pressed 4 times

varA = **0**

main:

add:

if pinC.1 = **1** then pause **100**

let varA = varA + **1**

pause **100**

endif

if varA>**4** then redledon

if pinC.1 = **0** then add

redledon:

high B.4

For varB = **0** to **10**

if pinC.0 = **1** then high B.1, B.2, B.3

pause **1000**

low B.1, B.2, B.3

pause **1000**

Endif

Next varB

goto main

stop

Display digits 1 to 9 / Select Case

main:

for cntA = **0** to **9**

select case cntA ' set output to leds high for each digit

case = **0** ' digit 0 to be displayed by leds

high B.1, B.2, B.3, B.4, B.5, B.6

low B.7

case = **1** ' digit 1 to be displayed by leds

high B.2, B.3

Low B.1, B.4, B.5, B.6, B.7

case = **2** ' digit 2 to be displayed by leds

high B.1, B.2, B.3, B.4, B.5, B.6, B.7

low B.3, B.6

case = **3** ' digit 3 to be displayed by leds

high B.1, B.2, B.3, B.4, B.7

low B.3, B.6

case = **4** ' digit 4 to be displayed by leds

high B.1, B.2, B.3, B.4, B.5, B.6, B.7

low B.1, B.4, B.5

case = **5** ' digit 5 to be displayed by leds

high B.1, B.3, B.4, B.6, B.7

low B.2,B.5

case = **6** ' digit 6 to be displayed by leds

high B.3, B.4, B.5, B.6, B.7

low B.1, B.2

case = **7** ' digit 7 to be displayed by leds

high B.1, B.2, B.3

low B.4, B.5, B.6, B.7

case = **8** ' digit 1 to be displayed by leds

high B.1, B.2, B.3, B.4, B.5, B.6, B.7

' all pins high so no low

case = **9** ' digit 9 to be displayed by leds

high B.1, B.2, B.3, B.6, B.7

low B.4, B.5

end select

pause **5000**

NEXT cntA

stop