MaxAir Technical – Input/Output Interfaces

In order to make it possible to install MaxAir on a range of SBCs (Single Board Computers) it has been necessary to employ an approach to input/output interfacing which is common across a large number of SBCs.

MaxAir runs on the Linux operating system and in general uses ‘Python’ libraries to control interfaces such as GPIO (General Purpose Input Output), SPI (Serial Peripheral Interface) and I2C (inter-integrated circuit).

MaxAir has been coded using the ‘Adafruit-Blinka’ library, which has been developed for SBCs from ‘CircuitPython’, a variant of MicroPython, a very small version of Python that can fit on a microcontroller.

‘Adafruit-Blinka’ has support for a large number of SBCs, including all the Raspberry Pi variants, most of the Orange Pi variants, many of the Pine64 boards, some of the Banana Pi boards and others. There is an active support community which is adding other boards as they become available.

‘Adafruit-Blinka’ works alongside the ‘Adafruit-PlatformDetect’ library, which is used to identify the board and chip type of the SBC.

The advantage of adopting this approach for MaxAir is that it can be deployed across a range of SBC with no code change required other than ‘pin mapping’ for GPIO functions.

GPIO Pin Mapping

MaxAir references GPIO pins using their physical pin number, for example for the Raspberry Pi, physical pin7 is GPIO4 and is named D4 by the ‘Adafruit-Blinka’ library. For the Orange Pi Zero 2 SBC, physical pin7 is GPIO-73 and is named PC9 by the ‘Adafruit-Blinka’ library. MaxAir will control the state of pin7 by using the number 7, but the ‘Adafruit-Blinka’ library will require the use of ‘D4’ in the case of the Raspberry Pi or ‘PC9’ in the case of the Orange Pi. This ‘pin mapping’ is achieved by the use of the Python script ‘Pin_Dict.py’.

Pin_Dict.py

This script uses ‘Adafruit-PlatformDetect’ to determine the SBC in use and then returns a mapping array, equating the physical pin number to the ‘Adafruit-Blinka’ pin name.

Example Python Code to Switch ON LED Connected to Physical Pin3 of Any Supported SBC

```
import board
import digitalio
from Pin_Dict import pindict
LED1GPIO      = '3'
led_1 = digitalio.DigitalInOut(getattr(board, pindict[LED1GPIO]))
led_1.direction = digitalio.Direction.OUTPUT
led_1.value = True
```