|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Item tested** | **Old Sensor** | **New Sensor** | **PI BCM GPIO** | **GPIO Pin #** | **Sensor Description** | **I2C Address** | **Python Command at prompt** | **Script content and notes** | **Reference** |
| Temp & humidity Sensor | [AM2320](https://www.aosong.com/en/products-41.html) | [DHT20](https://www.aosong.com/en/products-67.html) | SDA on GPIO2  SCL on GPIO3 | SDA on 3  SCL on 5 | I2C based Temperature and Humidity Sensor. |  | > python ths.py.  Ignore the first reading as it may be erroneous | Measure ambient temperature and humidity and display on screen. Ignore the first reading as it may be erroneous.  Use same python code to work with both AM2320 and DHT20 | Sample Code for [AM2320](https://learn.adafruit.com/adafruit-am2320-temperature-humidity-i2c-sensor/python-circuitpython) & [DHT20](https://wiki.seeedstudio.com/Grove-Temperature-Humidity-Sensor-DH20/#software) |
| Water pump Current sensing | [INA220AIDGSR](https://www.ti.com/store/ti/en/p/product/?p=INA220AIDGSR) |  | SDA on GPIO2  SCL on GPIO3 | SDA on 3  SCL on 5 | High- or Low-Side, Bidirectional Current and Power Monitor With Two- Wire Interface |  | > python current\_ina220.py. (Do this when motor is running and when motor is off to see the difference) | Script to measure current across shunt resistor to know when motor is ON/OFF. Pump off --> No current Pump on --> Current |  |
| Water Pump |  |  | EN\_Pump on PWM1/ GPIO13 | EN\_Pump on Pin 33 |  |  | > python waterpump.py | Switch pump on/off using PWM from Pi.  This will be hardware based PWM control. User configurable PWM frequency (typical 10K to 20KHz). Ramp up pump speed from 0% to 50% duty cycle  User configurable start & stop, PWM and duty cycle ramp up time. | Library for [pigpio](https://abyz.me.uk/rpi/pigpio/python.html#hardware_PWM)  You can use this for LED. Only difference will be that LED will need the Enable Signal generated from PI.  Command for hardware PWM  import pigpio  pi = pigpio.pi()  for i in range(1000000):  for j in range(0, 1000000, 100):  pi.hardware\_PWM(18, 400, j) |
| LED Column on/off |  |  | PWM\_LED on GPIO18/PWM0  EN\_LED on GPIO17 | PWM\_LED on Pin 12  EN\_LED on Pin 11 | LED Column On/ Off Control |  | >Python led\_onff.py | PWM\_LED frequency and duty cycle will control the lights and intensity. This will be hardware based PWM control. 95% duty cycle is full intensity and 0% duty cycle if no light  EN\_LED needs to be logic high to turn lights on / off.  User configurable start & stop, PWM and duty cycle ramp up time. | IC for Enable Signal  [UCC27523](https://www.ti.com/lit/ds/symlink/ucc27523.pdf?ts=1642387428630&ref_url=https%253A%252F%252Fwww.ti.com%252Fstore%252Fti%252Fen%252Fp%252Fproduct%252F%253Fp%253DUCC27523D) |
| LED Column Control |  |  | PWM\_LED on GPIO18/PWM0 & GPIO17 |  |  |  | > python fading.py | Script increments and then decreases the light intensity. Before you run this script, you will need to type following at command prompt. >sudo pigpiod |  |
| Camera | USB Camera |  | USB Port through USB Hub |  | Control and take image from two USB cameras connected to Pi through USB hub |  | > Python camera.py | Script to takes consecutive pictures from two cameras and store Image s Pi. Images are stored in the user defined folder.  User configurable image resolution, camera # and storage folder |  |

Diagram, schematic

Description automatically generated