Software for the CognIoT RadarPi is available on GitHub. The purpose of the software is to demonstrate the capabilities and provide some working software as a platform for starting project development.

# Radar Pi Sample Software

The RadarPi sample software is provided in a number of different programs, each showing a feature or capability of the RadarPi.

## Sample Software

The Sample Software provides some basic functions to demonstrate the basic capability of the RadarPi.

* d - Digital Detection (Filtered)
* D - Digital Detection (Raw)
* C – ADC Measurement

The difference between the filtered and raw measurement is signal processing. The filtered output contains additional processing for debounce and full signal detection. The program also configures for a default gain setting and sets the sample and hold pin for run.

To run: sudo ./sampleSoftware

## Analogue Measurement

The Analogue Measurement measures the voltage from the ADC processor and provides readings and functions for displaying the data.

* c – Read Voltage Continuously
* d – Create dataset of voltage & time for 30 seconds

The second of these options runs the ADC processor for 30 seconds and then displays a graph of voltage over time.

To run: sudo ./analogueMeasurement

## Digital Detection

The Digital Detection measures the signal input into the GPIO pins of the Pi as a frequency measurement.

* c – Read Frequency Continuously
* d – Create dataset of Frequency & time for 30 seconds

The second of these options runs the frequency measurement for 30 seconds and then displays a graph of frequency over time.

To run: sudo ./digitalDetection

Note: Also contained within DigitalDetection \ python directory is a python based program that performs a similar functions and graphing capabilities.

To run: sudo DigitalDisplay.py

## Gain Control

Contained within the RadarPi is a Gain processor which can set values between 0.25 and 157. When run, the program enables the user to set the gain to one of the following values.

|  |  |  |  |
| --- | --- | --- | --- |
| **G3:G0** | **GAIN (V/V)** | **SLEW RATE (V/μs)** | **SMALL-SIGNAL BANDWIDTH (MHz)** |
| 0000 | 1 | 2.90 | 2.15 |
| 0001 | 10 | 8.99 | 2.40 |
| 0010 | 20 | 8.70 | 1.95 |
| 0011 | 30 | 12.80 | 3.40 |
| 0100 | 40 | 12.50 | 2.15 |
| 0101 | 60 | 13.31 | 2.60 |
| 0110 | 80 | 12.15 | 1.91 |
| 0111 | 120 | 18.53 | 2.30 |
| 1000 | 157 | 16.49 | 1.78 |
| 1001 | 0.2 (VCC = 5V) | 2.86 | 1.95 |
| 0.25 (VCC = 3.3V) |
| 1010 | 1 | 2.90 | 2.15 |

By selecting the gain value required, the software automatically programs the correct value to achieve the required gain.

To run: sudo ./gainControl

## Test Program

This routine has been produced to enable the testing of the functionality of the board during manufacture. It requires external equipment to be plugged in and therefore should not be run.

# installation instructions

To install the software, follow these simple steps:-

1. Upgrade and Update the operating system
   1. sudo apt-get update
   2. sudo apt-get upgrade
2. Using Raspberry Pi Configuration utilities, configure the Pi for serial
   1. git clone <https://github.com/CognIot/utilities.git>
   2. cd utilities
   3. sudo ./cogniot-setup.sh
3. Download and Compile the Sample Software
   1. git clone https://github.com/BostinTechnology/RadarPi.git
   2. make
4. cd build

Al the sample software is contained within this directory. It is run by typing sudo ./<program name>

# Revision History

|  |  |  |
| --- | --- | --- |
| Version | Date | Comment |
| Draft |  | First version. |