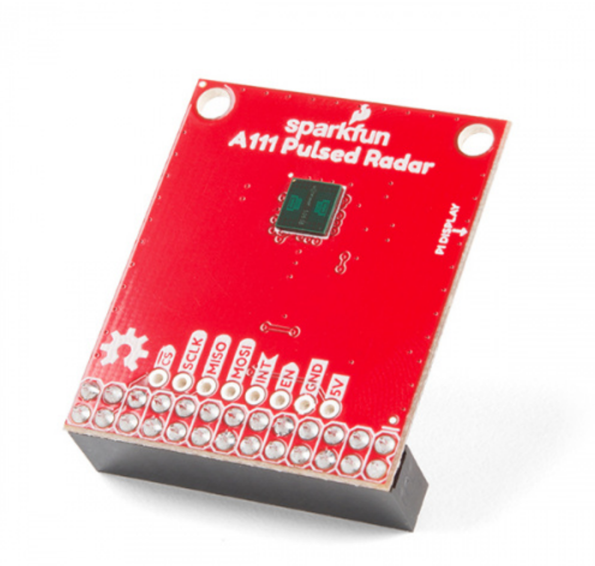
sparkfun
START SOMETHING

Getting Started with the A111 Pulsed Radar Sensor

Introduction

Heads up! If you are looking for the hookup guide for the SparkX version of the A111, you can check out the retired tutorial: Using the A111 Pulsed Radar Sensor with a Raspberry Pi. However, the older tutorial is outdated due to the SDK version used in the previous guide.

Does your project require high-precision, cutting-edge distance, speed, motion, and/or gesture sensing? We're not talking ultrasonic, or even infrared here, but 60GHz radar! Say hello to our tiny, pulsed radar friend the Acconeer A111!



SparkFun Pulsed Radar Breakout - A111
SEN-15577

Product Showcase: SparkFun A111 Pulsed Radar Breakout

The A111 is a single-chip solution for pulsed coherent radar (PCR) -- it comes complete with antennae and an SPI interface capable of speeds of up to 50MHz. Applications for PCR include distance-sensing, gesture, motion, and speed detection. The sensor can monitor one-or-more objects at distances of up to two meters.

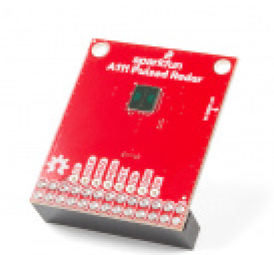
Our breakout board for the A111 includes a 1.8V regulator, voltage-level translation, and it breaks out all pins of the pulsed radar sensor to both 0.1-inch and Raspberry Pi-friendly headers.

Required Materials

> **Heads up!** The A111 was designed to work with an ARMv7 to work at a minimum. We were able to get the sensor working on the Raspberry Pi 3, 3B+, and 4.

To use the A111 you'll need either an ARMv7 or an ARM Cortex-M4 -- the closed-source SDK currently only supports these architectures. This tutorial will explain how to use the radar sensor with a **Raspberry Pi** -- a platform based on an architecture supported by the A111's SDK.

The A111 Breakout **includes a 20-pin, 2x10 female header**, which should mate to Raspberry Pi's of any generation. If you'd rather manually wire the A111 to your Raspberry Pi and about 9 male-to-female wires should do the trick.



SparkFun Pulsed Radar Breakout - A111
SEN-15577

Raspberry Pi 3 B+ Starter Kit
KIT-14644

Jumper Wires Premium 12" M/F Pack of 10
◉ PRT-09385

Optional Materials

You have several options when it comes to working with the Raspberry Pi. Most commonly, the Pi is used as a standalone computer, which requires a monitor, keyboard, and mouse (listed below). To save on costs, the Pi can also be used as a *headless* computer (without a monitor, keyboard, and mouse). This setup has a slightly more difficult learning curve, as you will need to use the *command-line interface* (CLI) from another computer.

Raspberry Pi LCD - 7" Touchscreen
◉ LCD-13733

SmartiPi Touch
◉ PRT-14059

Multimedia Wireless Keyboard
◉ WIG-14271

Suggested Reading

If you aren't familiar with the following concepts, we recommend checking out these tutorials before continuing. Using the Pi as a headless setup or with VNC can be useful when developing applications with the Pi.

Raspberry Pi SPI and I2C Tutorial

Learn how to use serial I2C and SPI buses on your Raspberry Pi using the wiringPi I/O library for C/C++ and spidev/smbus for Python.

Raspberry Pi 3 Starter Kit Hookup Guide

Guide for getting going with the Raspberry Pi 3 Model B and Raspberry Pi 3 Model B+ starter kit.

Headless Raspberry Pi Setup

Configure a Raspberry Pi without a keyboard, mouse, or monitor.

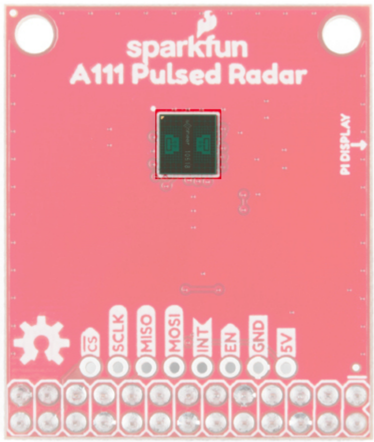
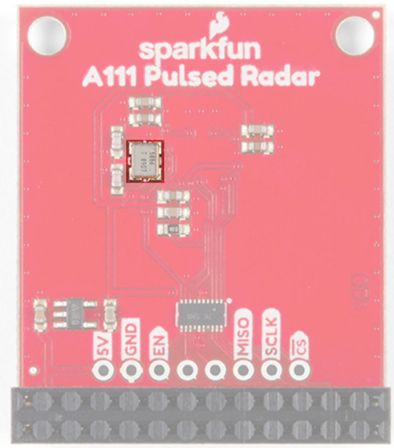
How to Use Remote Desktop on the Raspberry Pi with VNC

Use RealVNC to connect to your Raspberry Pi to control the graphical desktop remotely across the network.

Hardware Overview

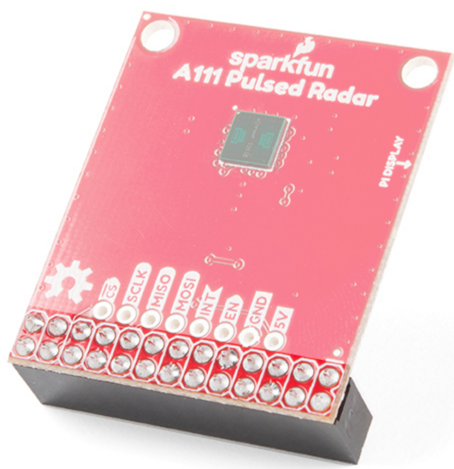
Sensor and Crystal

The breakout board is populated with the A111 IC and an external 26MHz crystal oscillator. The IC comes with two antennae and uses mmWave radio to sense objects via radar pulses.

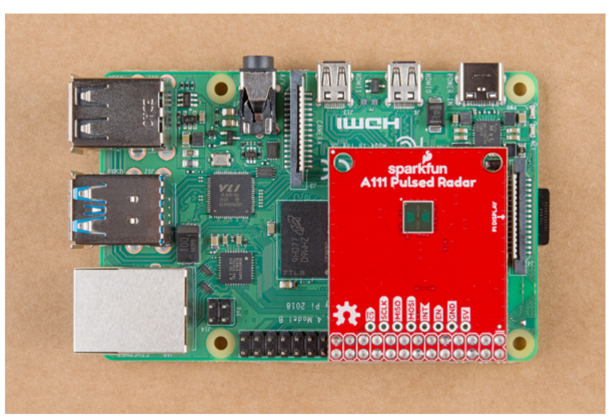
	
A111 Populated on Top	*Crystal Oscillator Populated on Back*

2x13 Female Header

To keep the size of the board small, there is a 2x13 female header soldered on the board to easily mate with a Raspberry Pi's standard header.

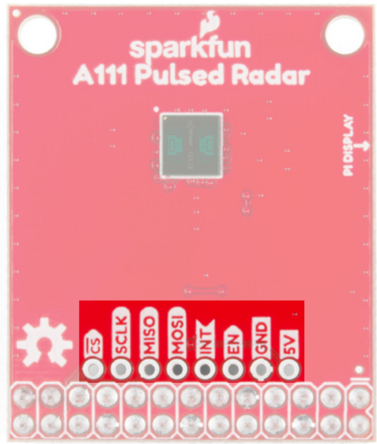


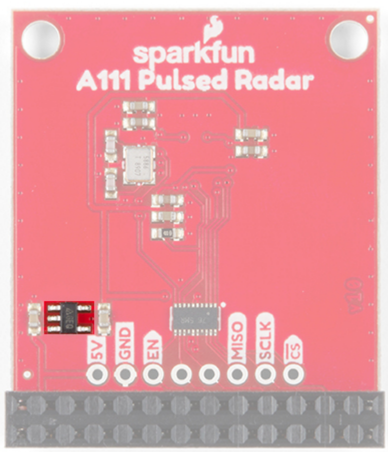
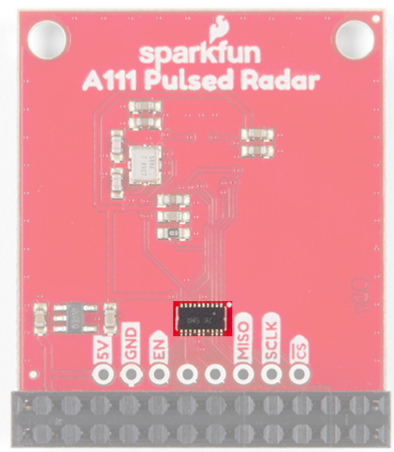
This should only cover the part of the 2x20 male headers on the Pi.



Additional Pins Broken Out

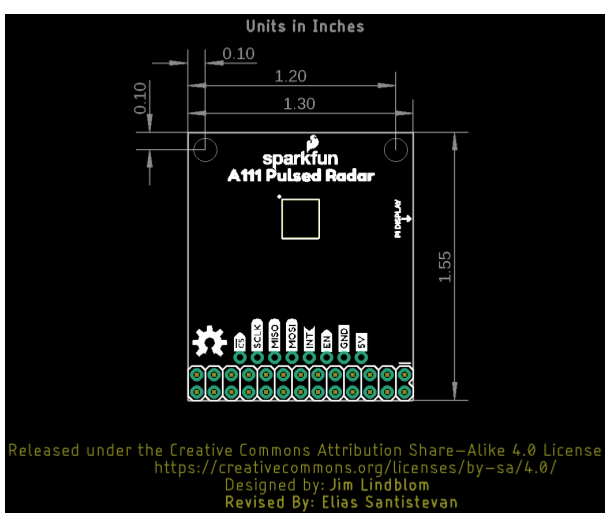
Just above the 26-Pin header, there are additional pins broken out for SPI, interrupt, enable, and power if you need access to the pins. The A111 IC requires an input voltage of **1.8V**, so a voltage regulator is populated on the back. Additionally, the I/O logic level is **1.8V**, so a logic converter is included to translate the signals.



	
Voltage Regulator	*Logic Level Converter*

Board Dimensions

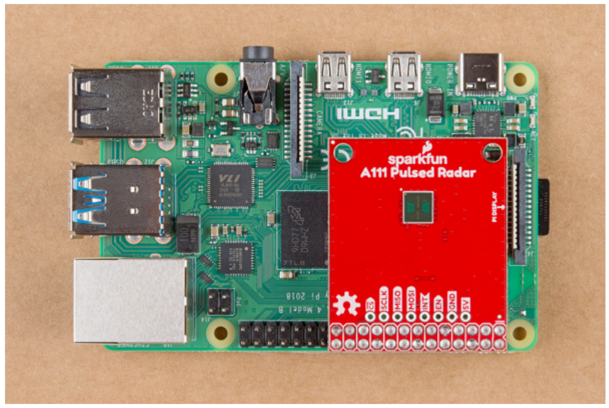
The overall board size is 1.30in x 1.55in. There are two additional mounting holes spaced 0.10 inches away from the edge of the board.



Hardware Assembly

The A111 Pulsed Radar Breakout is designed to sit directly on top of a Raspberry Pi. Again, it doesn't span all 40 (2x20) pins of a Raspberry Pi B+ (or later), but the 26-pin — 2x13 — header should be compatible with any Pi.

Connect the shield to a Raspberry Pi ensuring that the "Pi Display" text on the breakout matches up with the display header on your Pi. The sensor should be facing up after plugging it in.



A111 Breakout plugged into a Raspberry Pi.

Or, if you'd like to manually wire the breakout up to a Pi, here is the pin-out we'll use through the rest of this tutorial:

Breakout Pin	Raspberry Pi Pin Name	RasPi Hardware Pin Number
CS	SPI0 CS0	24
SCLK	SPI0 SCLK	23
MISO	SPI0 MISO	21
MOSI	SPI0 MOSI	19
INT	GPIO25	22
EN	GPIO27	13
GND	GND	6, 14, 20, etc.
VIN	**5V**	2, 4

⚡ **Input Voltage and Logic Levels:** "5V" should power the sensor, which can consume ~80mA.

When connecting to the additional pins broken out next to the Raspberry Pi's I/O, make sure to not accidentally connect anything higher than 3.3V since it is only 3.3V tolerant.

Configure Your Pi

Raspbian and SPI

This tutorial assumes you've already set up a Raspberry Pi with Raspbian. For help installing the Debian-based OS on your Pi, check out the docs on Raspberrypi.org. Or -- better yet! -- check out our Headless Raspberry Pi Setup tutorial.

You'll also need to **enable SPI** on your Pi. For help with that, check out our SPI on Pi tutorial.

We are going to assume you already have a Raspberry Pi up and running with Raspbian. We'll also assume that it is connected to the Internet. If not, check out our starter kits and tutorials on setting up a Raspberry Pi.

Raspberry Pi 3 Starter Kit Hookup Guide
APRIL 11, 2016
Guide for getting going with the Raspberry Pi 3 Model B and Raspberry Pi 3 Model B+ starter kit.

Make sure to update the image so that we have the latest distribution. Enter the following commands in the command line individually to update your image.

```
sudo apt-get update
sudo apt-get dist-upgrade
```

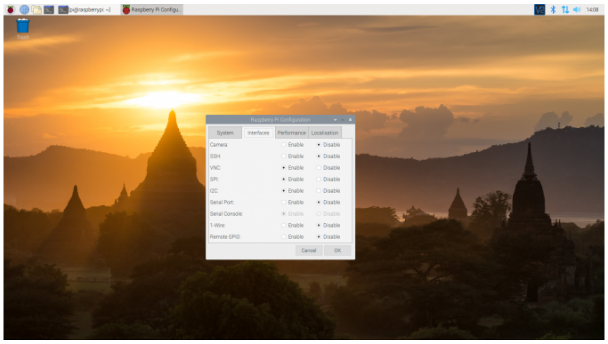
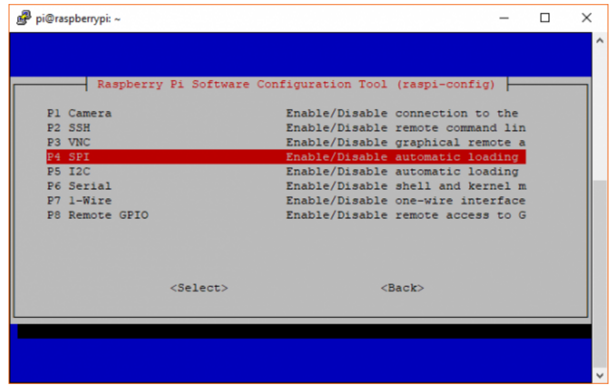
Note: sudo stands for "Super User Do", it is a way to give you superuser powers from the command line. Be careful whenever using `sudo`.

User Configuration Settings

Once you are set up, I highly recommend changing your default login and password: (**username**: pi, **password**: raspberry). The Raspberry Pi Configuration tool is a quick way to change your password as well as setup the network, language, keyboard, etc. There are two methods to adjust the settings. You can use the GUI by heading to the **Pi Start Menu** > **Preferences** > **Raspberry Pi Configuration** > **Interfaces**. For the old school heads, you can also use the **raspi-config** tool by typing the following command using the command line and then go through the menus to update your information.

```
sudo raspi-config
```

You'll want to enable the SPI pins using the tool to read sensor with SPI. If you are using hte GUI, simple select **Enable** and click on the OK button. If you are using the **raspi-config**, follow the prompts to finish setting up the SPI. For more information, check out our tutorial to set up the SPI pins on a Pi.

	
Enabling SPI via Desktop GUI	*raspi-config tool for SPI in the Terminal*

You'll need to restart your Pi for the changes to take effect.

Get the SDK

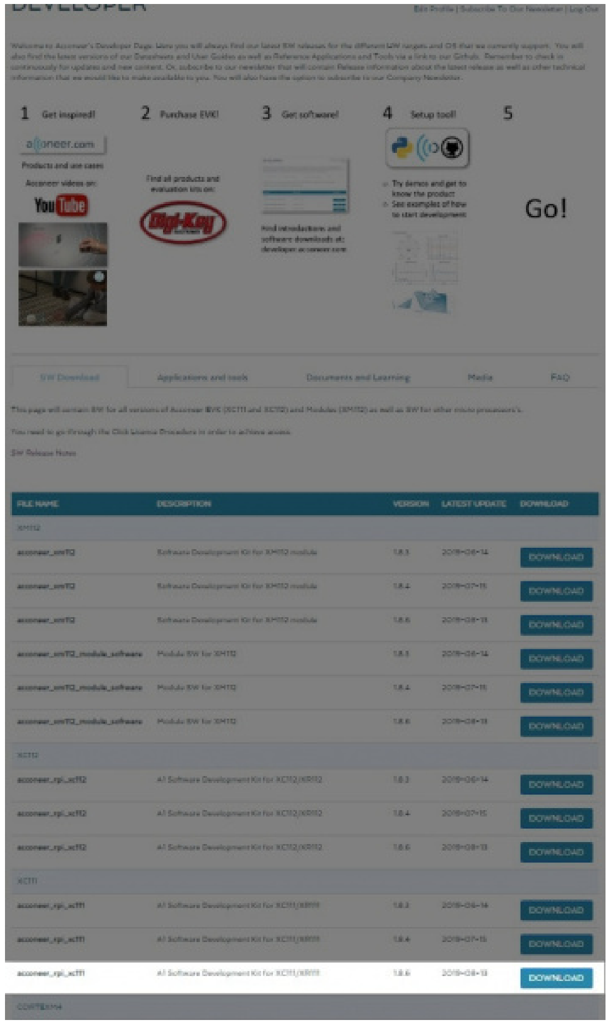
The software development kit (SDK) for the A111 is, unfortunately, locked behind a closed source blob that currently only supports Cortex-M4 and ARMv7 platforms. However the folks over at Acconeer have agreed to host the latest version of their SDK with our example code within SparkFun's Pulsed Radar Example Code repository on Github. To download the SDK from Acconeer, visit Acconeer's "Developer" page on your Raspberry Pi's browser.

VISIT ACCONEER'S "DEVELOPER" FOR THE SDK

You'll need an account in order to have access to the files. Click on the link at the bottom of the page to create one or log in. Or click on the shortcuts below.

SIGN UP WITH ACCONEER **LOG IN WITH ACCONEER**

Once you are logged in, the "**SW Download**" tab will include additional links to download the SDK. Scroll down the webpage to the section for the XC111. Acconeer is constantly updating and improving the software. We used **v1.8.6** of the software when writing this tutorial so the version may be different from the version that you download. Click on the **Download** button for **v1.8.6**.



Download the SDK from Acconeer.

Read through the license, agree, and click on the **Download acconeer_rpi_xc111** button for the software. Depending on your browser, you may need to refresh the page and repeat the directions before the button is active.

Once downloaded, head to directory where you downloaded the ZIP. You can use the GUI or you can use the terminal to unzip the SDK using the following commands (included are commands to install unzip). Just make sure to update the file name with the version that you downloaded. Assuming that you are in the **Downloads** folder, you will be using the following command in the Terminal.

```
unzip acconeer_rpi_xc111_v1_8_6.zip -d a111
```

Note: Make sure you replace the acconeer ZIP file name with that of your downloaded SDK version.

Then `cd` to the "**a111/rpi_xc111/...**" directory to prepare to build the example software. Depending on the version that you downloaded, the path might be slightly different. Make sure to use the `ls` to verify the contents of the unzipped **a111** folder if you have issues navigating through the directory.

SDK Overview

The A111 SDK includes source code, archived libraries, include files, and documentation for using the A111 pulsed radar sensor. Here's a quick overview of what's included with the SDK:

- **doc** — Doxygen-generated documentation for the A111 API and source code.
- **include** — Header and API files which describe how to interact with the pre-compiled A111 libraries.
- **lib** — Pre-compiled A111 static archives. API for these files are provided in the "include" directory.
- **out** — Compiled board and example object and executable files.
- **rule** — Recursive Makefile rules for board and example files.
- **source** — C source files for custom boards and example applications.
- **makefile** — Top-level makefile. Recursively calls files in the "rule" directory to build example and board files.

Adding Custom Example and Board Files

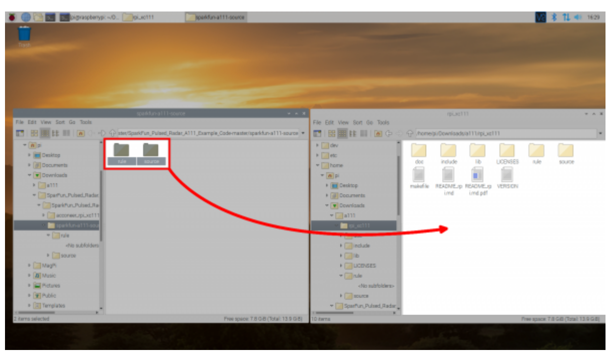
The SparkFun A111 Breakout's default pins will not work with those of the SDK's examples. To build and run an example with this board, we have an example board definition, make scripts, and example applications. Click the button below to download these files from the GitHub repo. The code currently works for v1.8.6.

DOWNLOAD THE SPARKFUN A111 EXAMPLE CODE (ZIP)

Once downloaded, head to the folder where you downloaded the example code and unzip the files. You can use the GUI to extract by right clicking and selecting **Extract Here**. Or you can use the following command if you are using a Terminal. Make sure to use the `cd` so that you are in the same folder as the ZIP file.

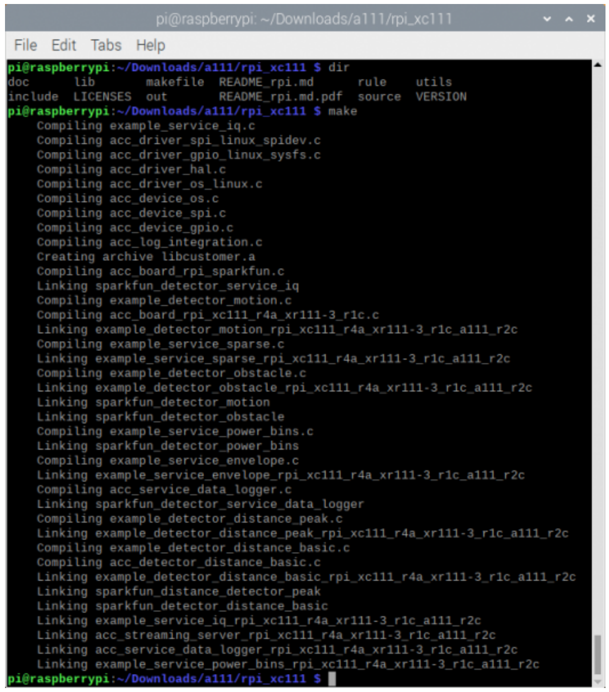
```
unzip SparkFun_Pulsed_Radar_A111_Example_Code-master.zip -d SparFun_Pulsed_Radar_A111_Example_Code-master
```

Once extracted move the files provided from the **../SparkFun_Pulsed_Radar_A111_Example_Code-master/sparkfun-a111-source** to **...a111/rpi_xc111**. We recommend opening a separate window using the GUI to drag and drop the files into the file path. This will add the SparkFun examples into the respective paths. You'll get a window that pops up. Click the **Overwrite** button for each window that pops up to add the files into their respective folders.



Build and Run the Test Sketch

In the terminal, executing the make file — and it's recursive dependencies — should build all of the examples you may use with the **A111**. To build all board and example files, navigate to the SDK's top-level directory (in this case, it should be similar to **...Downloads/a111/rpi_xc111**) where your **makefile** is located and type `make` in the terminal. You should see something similar to the output below.



Troubleshooting

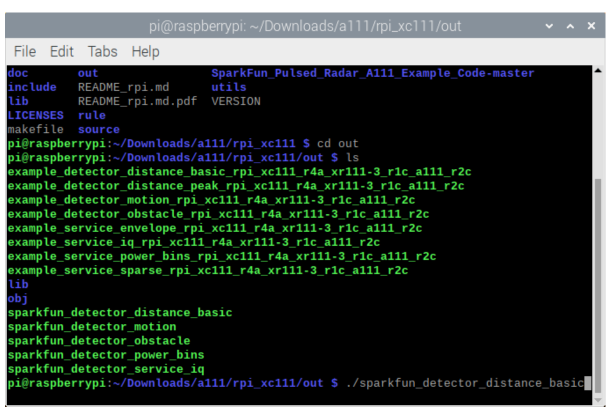
If you have any trouble building the board and example files, ensure that you have gcc and make packages installed. (E.g. `apt-get install make gcc`)

Running the Example Applications

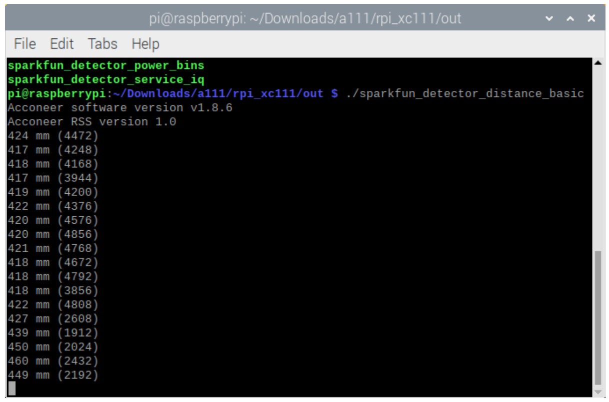
Once compiled you can run the example applications. Navigate to the **.../out** folder with the `cd` in the terminal window. You can run the example by clicking on the executable or typing the following command. We recommend typing the command in the terminal to view the data. A few of the applications will automatically close the window after taking sensor data if you choose to click on the executable.

```
./sparkfun_detector_distance_basic
```

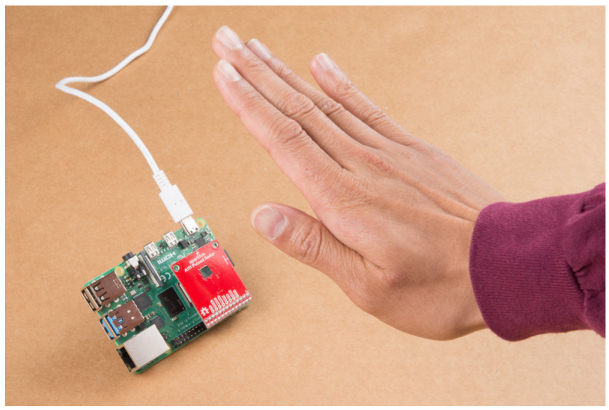
Once you type (or paste the command in the Terminal), hit `Enter` on your keyboard to execute the command.



This will run our modified distance-detector example application. For simplicity, it will show the distance of an object that is in front of the antennae.



By placing an object in front of the sensor, the distance should vary based on what the A111 can detect. The application will continue running in the terminal window until `CTRL + C` is pressed.



Hand placed in front the A111 with the Pi remotely connected via VNC..

Running Other Examples

In addition to the distance detector basic, the SDK includes a few extra examples. To execute these files, run them from the **out** directory:

- **sparkfun_detector_motion** — If there is an object moving in front of the sensor, it will output a true. Otherwise, the output is false.
- **sparkfun_detector_obstacle** — Checks to see if there is an object in the way.
- **sparkfun_detector_service_iq** — Advanced version of the envelope service, which includes phase information for very small variations in distance. The output will be in polar coordinates.
- **sparkfun_detector_power_bins** — Demonstrates how to use A111 "power bins," which are still, kind of, a mystery to us...
- **sparkfun_distance_detector_peak** — This application will begin by calculating raw peak-distances, with a maximum of ten reflections. The example will briefly estimate minimum and maximum threshold.

Make sure to read through the comments in the code (located in the **source** folder), datasheet, user guide, and associated documents from Acconeer for more information before writing your own custom code for the A111.

Resources and Going Further

Now that you've successfully got your A111 pulsed radar sensor up and running, it's time to incorporate it into your own project! For more information, check out the resources below:

- Schematic (PDF)
- Eagle Files (ZIP)
- Board Dimensions

- Acconeer
  - Developer - SW Download, datasheet, user guide, FAQ
  - Read the Docs: Setting Up Your Raspberry PI EVK - Documentation for the Python examples that might be useful in visualizing the data taken with the A111 examples.
- GitHub Repos
  - Acconeer Example Code
  - SparkFun Example Code
  - Hardware Repo
- SFE Showcase

Need some inspiration for your next project? Check out some of these related tutorials:

Getting Started with Walabot

See through walls, track objects, monitor breathing patterns, and more using the power of radio frequency with the Walabot! In this tutorial, we will explore Walabot’s features using the Software Demo Kit (SDK) on Windows and the Application Programming Interface (API) on Linux-based distributions for embedded projects.

Qwiic Kit for Raspberry Pi Hookup Guide

Get started with the CCS811, BME280, VCNL4040, and microOLED via I2C using the Qwiic system and Python on a Raspberry Pi! Take sensor readings from the enviroment and display them on the microOLED, serial terminal, or the cloud with Cayenne!