

# PROGRAM\_NAME 0.1

User manual

David Buckingham

Eric Tytell

Cassandra Donatelli

Margo

## Contents

0.1	Copyright and License	3
0.2	Introduction	3
0.3	Hardware	3
0.3.1	PC	3
0.3.2	Arduino	3
0.3.3	IMUs	3
0.3.4	Wiring the Arduino	4
0.4	Installation	4
0.4.1	Arduino code	5
0.5	Collecting data	5
0.5.1	Data buffer	5
0.5.2	Trigger	5
0.6	Processing data	7
0.7	Saving and loading	7
0.7.1	.csv file format	7
0.8	Settings	7
0.9	Troubleshooting	7
0.9.1	Error messages	7

## 0.1 Copyright and License

PROGRAM\_NAME: a tool for collecting IMU measurements. Copyright (C) 2017 Author(s)??????/

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see [<http://www.gnu.org/licenses/>](http://www.gnu.org/licenses/).

## 0.2 Introduction

## 0.3 Hardware

PROGRAM\_NAME interconnects three pieces of computational hardware:

1. a PC
2. an Arduino
3. between 1 and 3 Inertial Measurement Units (IMUs)

### 0.3.1 PC

PROGRAM\_NAME has been tested on PCs running Linux (Linux Mint 18 Sarah), macOS (Need to test this and put version), and Windows (Version??). It uses multi-threading and will perform best if multiple processing units (cores) are available.

### 0.3.2 Arduino

PROGRAM\_NAME has been tested with an Arduino UNO. It is expected to work on other models having the same clock speed (16MHz) and sufficient storage.

### 0.3.3 IMUs

The mpu9250 by InvenSense is a nine-axis (gyroscope, accelerometer, compass) motion tracking device. For testing purposes, or if the added size and weight are acceptable for your application, a mpu9250 mounted on a circuit-board can be used with PROGRAM\_NAME.

For applications requiring minimum package size and weight, it is possible to wire the mpu9250 directly to the arduino. A separate manual, **name**, documents the procedure we have used to prepare the mpu9250 for use with PROGRAM\_NAME.

description	label	color	pin
IMU 1 chip select	NCS / CS	white	8
IMU 2 chip select	NCS / CS	white	9
IMU 3 chip select	NCS / CS	white	10
data from Arduino to IMU	MOSI / SDA	green	11
data from IMU to Arduino	MISO / SDO	blue	12
clock	SCL / CLK	yellow	13
power		red	3.3V
ground		black	GND
trigger			4

### 0.3.4 Wiring the Arduino

Wire each IMU to the Arduino. It may be necessary to use a breadboard, especially if multiple IMUs are used. Table 0.3.4 summarizes the wiring process. The Arduino communicates with the IMUs using the Serial Peripheral Interface bus (SPI) protocol. Each IMU needs a separate *chip select* line, but all IMUs share the other lines. Pins 8, 9, and 10 are used for *chip select* lines for up to three IMUs. Pin 11 is for data traveling from the Arduino to the IMUs, or Master-Out, Slave-In (MOSI). Pin 12 is for data traveling from the IMUs to the Arduino, or Master-In, Slave Out (MISO). Pin 13 is for a clock signal, used to regulate timing of the communication protocol.

If using an optional trigger, wire it to Pin 4.

In addition, each IMU should be connected to 3.3V power (available on the Arduino) and each IMU and the trigger should be connected to Ground.

Connect the arduino to the PC with a USB cable.

To ensure adequate power for the Arduino and IMUs, it is recommended to power the Arduino with an external power source instead of relying on the USB port.

## 0.4 Installation

PROGRAM\_NAME requires Python3.

We recommend using pip to install PROGRAM\_NAME: <https://pypi.python.org/pypi/pip>

```
# pip install --upgrade pip
# pip install \name
```

Pip will automatically install any of the following dependencies if needed:

- h5py
- numpy
- pyqtgraph
- pyserial

- pyqt5

### 0.4.1 Arduino code

#### Arduino IDE

While there are many tools for installing program code onto the Arduino, we have tested PROGRAM\_NAME using the Arduino IDE.

The software can be downloaded from: [urlhttps://www.arduino.cc/en/Main/Software](https://www.arduino.cc/en/Main/Software)

If you'r operating system has a package management system, it might be able to install the Arduino IDE. For example, on Debian systems, use:

```
# apt-get install arduino-core
```

#### Install PROGRAM\_NAME on the Arduino

Install the file am\_tx/am\_tx.ino on the Arduino. This can be accomplished using the Arduino IDE graphical user interface.

Alternatively, using Arduino IDE version 1.5.0 or later, the Arduino can be programmed directly from the command line:

```
# arduino --upload am_tx/am_tx.ino
```

For more information about the Arduino command line interface, see: <https://github.com/arduino/Arduino/blob/master/build/shared/manpage.adoc>

## 0.5 Collecting data

To begin collecting data, press the “record” button.

### 0.5.1 Data buffer

The “data buffer length (# samples)” slider adjusts the size of the data buffer...

### 0.5.2 Trigger

Optionally, a trigger attached to the Arduino (Section 0.3.4) can be used to stop recording. If the “use trigger” checkbox is not checked, the trigger is ignored. Otherwise, if an “active” trigger is detected, recording will stop, i.e., the same effect as pressing the “stop” button during recording. If you attempt to begin recording while the “use trigger” checkbox is checked and the trigger is “active”, the recording end as soon as it starts with zero data stored.

If the “invert trigger” checkbox is not checked, the trigger is considered “active” when the associated Arduino pin is set high. If the “invert trigger” checkbox is checked, the trigger is considered “active” when the associated Arduino pin is set low.

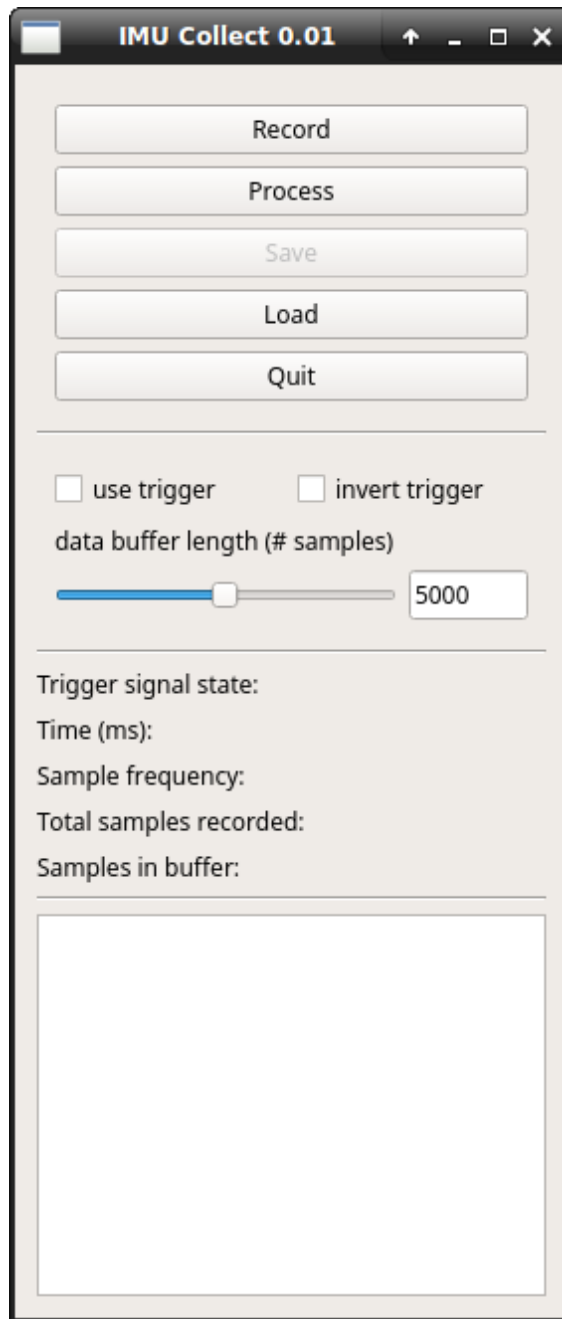


Figure 1: Control panel

If there is nothing setting the current on the Arduino pin associated with the trigger (i.e. if no trigger is connected), then the value of the pin is undefined. Thus, to ensure reliable behavior, the “use trigger” checkbox should not be checked unless there is a trigger connected to the Arduino.

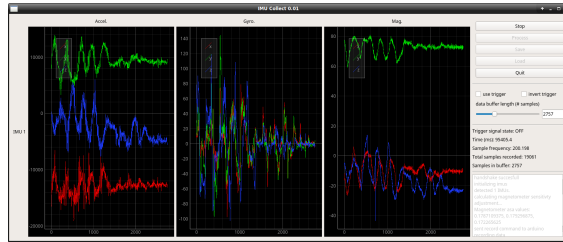


Figure 2: Collecting data from a single IMU

## 0.6 Processing data

## 0.7 Saving and loading

### 0.7.1 .csv file format

PROGRAM\_NAME uses the CSV module in the Python Standard Library with default formatting parameters:

<https://docs.python.org/3/library/csv.html>

## 0.8 Settings

## 0.9 Troubleshooting

### 0.9.1 Error messages

#### invalid csv file

The program attempted to read a .csv file (Section 0.7.1), but the format of the data in the file was not valid.

#### invalid file type: ...

#### rx failed, no data read from serial

#### no Arduino found

#### failed to create connection

#### handshake failed

**no connection, aborting**

**unable to determine number of IMUs, aborting**

The program failed to determine how many IMUs are attached to the Arduino. After the Arduino is initialized, it attempts to determine the number of IMUs by sending a WHOAMI request while signaling each of the three legal chip select pins (Section [0.3.4](#)). It then sends a message to the PC reporting the number of IMUs detected. This error is reported if the PC program sends a command to the Arduino to initialize, but does not receive a message reporting the number of IMUs detected. Make sure that any IMUs are correctly wired to the Arduino (Section [0.3.4](#)) and that the correct code is installed on the Arduino (Section [0.4.1](#)), and try resetting the Arduino.

**no IMUs detected, aborting**

**ASA read failed, using 1 adjustment**

**Sample packet too short: ...**