uRAD

User Manual

Arduino version

Contents

About uRAD 1

Congratulations on purchasing uRAD

⚠ *WARNING:* To avoid any injury or damage, read all operating instructions in this guide and, specially, the safety and warranty information in “Chapter 5: Safety and Handling” and “Chapter 6: Product Warranty”, before using uRAD.

You are close to transform your Arduino board into a functional microwave radar. Through this manual, you will learn how to use uRAD for an unlimited number of applications related with measuring distance and velocity of any element of your surrounding world. uRAD is conceived as an evaluation platform which brings to you radar detecting technology with a simple but high-performance device whose specifications are at the same level as professional radars.

uRAD is an Arduino shield. Therefore, it only works together with an Arduino compatible board. The boards you can use are those that have the usual pins disposition: Arduino Uno, Zero, Leonardo, 101, Mega, Due, M0, Yun and Ethernet, among others.

Additional Information

Released versions:	uRAD v1.0	15/07/2018
Purchasing:	www.uRAD.es/en/arduino	
Technical specifications:	www.uRAD.es/en	
Software download:	www.uRAD.es/en/mi-cuenta/downloads	
Contact:	contact@uRAD.es	

uRAD Basics

2

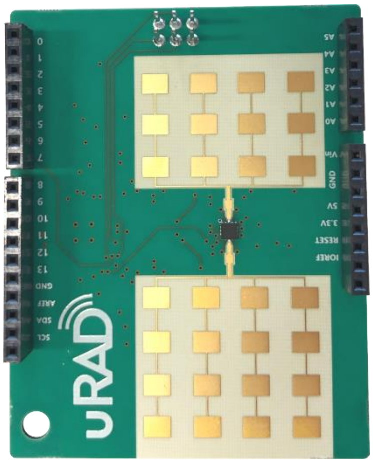
Read this chapter to learn about the features of uRAD, how to use it, and more.

uRAD is a tiny device which includes outstanding radar technology. However, its operation is simple to understand and easy to control.

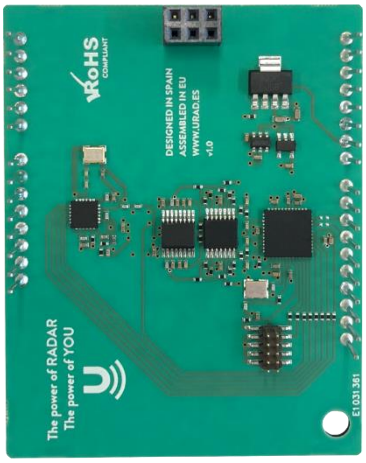
Hardware Description

uRAD is fabricated in planar technology over a high-performance substrate. It is a multilayer printed circuit board with many integrated chips on it.

- The top layer includes the uRAD's core, a 24 GHz transceiver, and the transmitter and receiver antennas. The microwaves are emitted perpendicular to this layer, so you should point this face to the direction of interest. Be careful because it does not emit backwards.
- The bottom layer consists of power supply and signal processing elements, which are managed by a powerful microcontroller.
- Four different connectors are placed in both longest sides, which are the interface with Arduino boards.



Top view



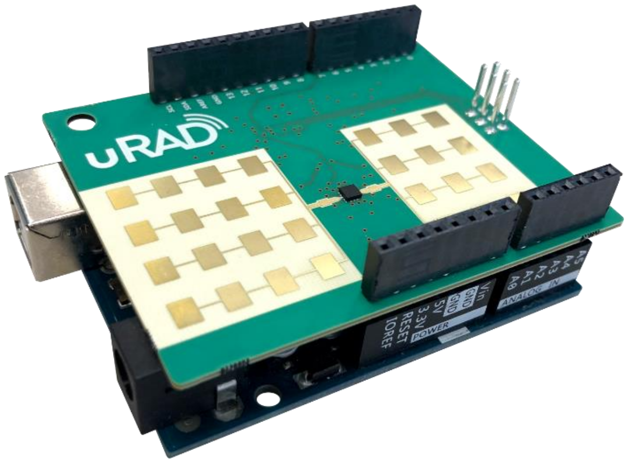
Bottom view

Side view

Assemble with Arduino

Assembling uRAD into your compatible Arduino board is as simple as inserting the male contact pins of uRAD side connectors into the female Arduino connectors. After assembling them, the visible part of uRAD is the top layer or antenna side.

WARNING: Male pins must not be inserted completely. Push uRAD up to there is enough contact between the connectors, avoiding any bend in the board.



uRAD is powered by the 5V power pin of Arduino and they also share the GND power pins. Therefore, for powering uRAD you only need to power up your Arduino as usual. Its power consumption is 0.85 W (current = 170 mA). Moreover, uRAD shares 2 digital pins with Arduino for communication and ON/OFF control of uRAD besides the ICSP connector.

- Digital pin 6: logic HIGH/LOW on this pin, switch ON/OFF uRAD, respectively.
- Digital pin 7: Slave Select pin for SPI communication protocol.
- ICSP connector: transmission and reception of data between uRAD and Arduino.

WARNING: Digital pins 6, and 7 are reserved for uRAD–Arduino interaction and therefore are not available for any other purpose. **DO NOT USE THEM**. The remaining pins, including 5V, GND and ICSP, can be used in integration with additional projects.

WARNING: Do NOT cover the antennas with any metallic or electronic element, nor electromagnetic absorber. Working with the provided case do not affect its performance. Most thin plastics are practically invisible to emission.

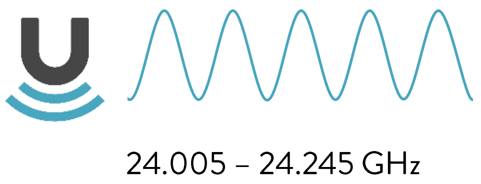
Main Features

Basically, a radar is a device that detects elements which are in its action radius. It works emitting an electromagnetic wave to the air. This wave is reflected by the target and comes back to the radar. The shape of the going and coming wave as well as the radar architecture determines the type of the radar.

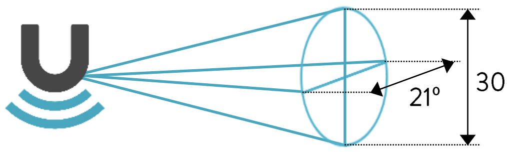
uRAD can detect up to 5 different elements that are in its field of view. Information about distance to the target and the relative radial velocity to it, is provided. Moreover, the amount of reflected power is also given, that serves as an estimation of the object size or the detection level. Whether not enough power is reflected, the element is not detected. Therefore, very small or far targets will not be seen.

The main features of uRAD are:

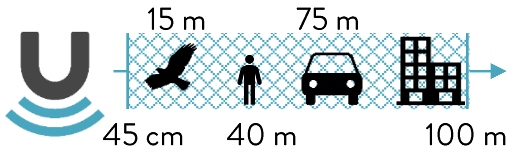
Frequency Bandwidth



Field of View

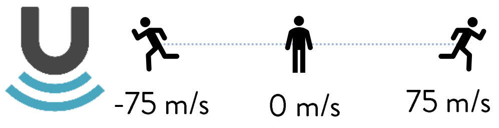


Distance Range



Accuracy = max. ±4 cm or 0.3%

Velocity Range



Accuracy = ±0.05 m/s

Distance/velocity range and accuracy depend on the configuration setup.

> *WARNING:* Detected velocity is only radial velocity: the component of the target's velocity that points in the direction of the line connecting the object and the radar.

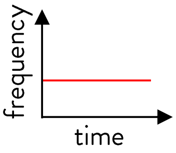
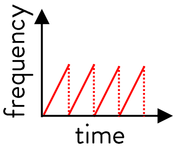
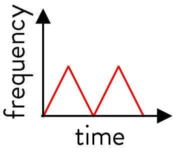
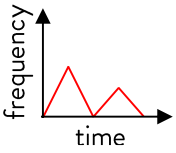
Additional information regarding technical features can be found in www.urad.es.

Configuration Parameters

You must enter 8 parameters for configuring your uRAD shield. In Chapter 3, we will show you how to send this configuration from Arduino to uRAD. Here, you will learn which configuration is more convenient for your application.

1. Mode

There are four operation modes that corresponds to 4 different transmitted waveforms: continuous wave (CW), sawtooth, triangular and dual rate. In CW mode a single frequency is transmitted. This is the most common mode used by doppler radars. The rest of the modes, a frequency ramp is transmitted, also called frequency modulated continuous wave (FMCW). Each mode has their advantages and disadvantages and depending on the application, you will select one or another. Next table summarizes the main features of each mode.

Mode	1	2	3	4
Name	CW	Sawtooth	Triangular	Dual Rate
Waveform				
Measured parameters	Velocity	Distance	Distance Velocity	Distance[1] Velocity[1]
Movement detector	YES	YES	YES	YES
Distance range (m)	0 to 60	0.45 to 100	0.45 to 100	0.45 to 75
Distance accuracy (m)	-	Max: ±0.3% ±0.04	Max: ±0.3% ±0.04	Max: ±0.3% ±0.04
Distance resolution[2] (m)	-	1.5	Different[3] velocity or 1.5	Different[3] velocity or 1.5
Velocity range (m/s)	0 to ±75	-	0 to ±75	0 to ±75
Velocity accuracy (m/s)	±0.05	-	±0.25	±0.25

Velocity resolution[2] (m/s)	3	-	Different[3] distance or 3	Different[3] distance or 3
Update rate max[4] (samples/second)	38	21	13	7

- [1] The fact that the dual rate mode uses two different triangular ramps consecutively, provide enhanced results due to ghost targets reduction. **This mode is especially useful in multi-target scenarios.**
- [2] Distance and velocity resolution indicates de minimum distance or velocity that two targets must be separated to be discerned as a single target each one.
- [3] In mode 3 and 4, uRAD can discern two targets at exactly the same distance but different velocity, and vice versa, respectively.
- [4] Update rate is maximum when the number of samples is minimum.

2. f0

It is the operation frequency in CW mode or the ramp start frequency in the others. Since uRAD is configured to operate between 24.005 and 24.245 GHz, you can select f0 from 5 to 245 in CW mode or from 5 to 195 in the others (minimum frequency sweep allowed in ramp modes is 50 MHz).

f0	Mode 1	Mode 2, 3, 4
Range value	5 to 245	5 to 195

3. BW

It is the operation bandwidth in ramp modes (modes 2, 3, 4). In other words, the frequency sweep which is varied in every ramp. Depending on the f0 introduced, there will be a larger or lower BW available to select. Minimum value is 50 and in the case you try to introduce a value higher than the available BW, uRAD will select the maximum BW allowed.

$$BW_{max} = 245 - f0$$

BW is a very relevant parameter because defines the accuracy of the system. The higher the BW, the better the accuracy. Moreover, selecting higher BW makes uRAD more capable to distinguish between targets that are very close to each other.

BW	Mode 1	Mode 2, 3, 4
Range value	-	50 to (245 – f0)

Because mode 1 is mono-frequency, the BW here has no sense and the value introduced causes no effect on the configuration.

ADVICE: Although minimum BW is 50, we recommend using the maximum BW available, or at least 150, unless your specific application requires lower BW.

ADVICE: If you have more than one uRAD module detecting the same area, select different f0 and BW for each one to reduce mutual interference.

4. Ns

It defines the number of samples that uRAD takes of the reflected wave to calculate distance, velocity, etc. This parameter is even more important in modes 2, 3, 4 because it also defines the ramps duration and therefore the update rate.

Ns	Mode 1	Mode 2	Mode3	Mode 4
Range value	50 to 200	50 to 200	50 to 200	50 to 200
Update rate [samples/second]	38 to 21	21 to 15	13 to 9	7 to 5

ADVICE: In mode 1, selecting higher Ns makes uRAD able to better discern between targets that have very similar velocity.

It seems that it is always better to select the lowest Ns to have the best update rate. However, the relation between BW and Ns also determines the theoretic maximum distance that uRAD can see.

$$Distance_{max} = 75 \times \frac{Ns}{BW}$$

Therefore, the lower maximum distance (Ns = 50, BW = 240) is only 15.625 meters whereas the higher maximum distance (Ns = 200, BW = 50) is 300 meters.

WARNING: although the theoretical maximum distance can be up to 300 meters, uRAD does not emit enough power to see targets at 300 m. So that, the actual maximum detection distance is around 100 meters.

5. Ntar

It is the maximum number of targets to detect. A maximum of 5 targets can be selected. If more than 5 elements are in the field of view, uRAD gives you the information of the 5 most significant ones that reflect more power.

Ntar	Mode 1, 2, 3, 4
Range value	1 to 5

6. Rmax / Vmax

Rmax is the maximum distance where the targets will be searched. Rmax is independent of the theoretical maximum distance, which defines the detection range. With Rmax you establish the length of the zone you want to detect.

For example, imagine that you fix BW = Ns = 100 that gives you a theoretic maximum distance equal to 75 m. If you define Ntar = 3 and Rmax = 20, uRAD will give you de information of 3 targets that are located between 0 and 20 meters, independently that uRAD detects more relevant targets in the range from 20 to 75 meters.

On the other hand, in mode 1, uRAD is not able to obtain distances. Therefore, Rmax here means Vmax, the maximum velocity range where the targets will be searched.

Rmax / Vmax	Mode 1	Mode 2, 3, 4
Range value	0 to 75	1 to 100

If you select Rmax = 100 in mode 2, 3 or 4, uRAD will search targets in the range defined by the maximum distance formula. So you can detect targets farther than 100 meters.

7. MTI

It activates or desactivates the Moving Target Indication (MTI) mode. **In this working mode, all static objects are ignored, and only the information of targets that are in movement respect to uRAD is provided.** This feature is only usually available in high performance radars due to the complexity in the data processing. Because of that, it is necessary to define a suitable sensitivity for each scenario.

A value equal to 0 indicates that MTI mode is disabled, and therefore, the information of ALL targets, static or not, is given. A value equal to 1 indicates that the mode is activated and therefore, the static targets are ignored and their information is omitted.

MTI	Modo 1	Modo 2, 3, 4
Range value	-	0 (disabled), 1 (activated)

In mode 1, MTI has no relevance because, by default, this mode is only for moving targets.

ADVICE: This mode can be very useful indoor, for instance, where there may be lot of no relevant static targets and it is only desired to measure the moving ones.

8. Mth

It defines the sensitivity of uRAD when it is working as a movement detector. As you will see in the next section, uRAD alerts you when it detects that some target is moving in its detection area, whenever you want this alert.

With Mth, you can define up to 4 detection thresholds. This threshold is defined as function of target reflectivity which is proportional to the size of the target and inversely proportional to its distance (bigger and closer targets reflect more).

Mth = 4 makes uRAD extremely sensitive to any reflectivity, whereas Mth =1 means that only very reflective targets active the alert.

Mth	Mode 1, 2, 3, 4
Range value	1 (low) to 4 (high) sensitivity

ADVICE: Try several values of Mth to find the value that better fits your particular scenario. In this way, you will reject undesired alarms.

WARNING: MTI and Mth are independent. You can define MTI = 0 for receiving the information of both static and moving targets, and at the same time, obtaining the movement alert with Mth.

Detected Information

uRAD provide you with complete information of its detection range:

- Distance: returns the distance from uRAD to each detected target.
- Velocity: returns the radial relative velocity between uRAD and each detected target.
- SNR: returns the Signal to Noise Ratio of each detected target. This gives you an idea of the amount of reflected signal of each target, and therefore the size and reflectivity of the target. Technically speaking, it is the difference in magnitude between the reflected signal due to the target and the noise floor due to the whole system. **SNR will hardly exceed a value of 40, in any scenario.**
- Movement: returns TRUE or FALSE if movement of any target IS or IS NOT detected.
- I, Q: returns the total reflected signal decomposed in two arrays with the In-phase and Quadrature components, for advanced data signal processing.

Available returned information depends on the configuration mode.

	Mode	Unit
Distance	2, 3, 4	meters
Velocity	1, 3, 4	meters/seconds
SNR	All	dB
Movement	All	TRUE/FALSE
I, Q	All	Arbitrary units from 0 to 4095

Programming 3

In this chapter, you will learn how to program your Arduino in order to use uRAD.

Just little programming knowledge is necessary for controlling uRAD with Arduino. Here you will find the basic pieces of code for programming the interaction between uRAD and Arduino, as well as a full example of use.

⚠ *WARNING:* Along this chapter, reserved words appear in red color, while green words are generic names that can be chosen by the user.

Getting Started with Arduino

Arduino is an open-source electronic platform based on easy-to-use hardware and software. The Arduino Software (IDE) allows you to write programs and upload them to your board.

You can download the Arduino software from www.arduino.cc/en/Main/Software for Windows, Mac or Linux.

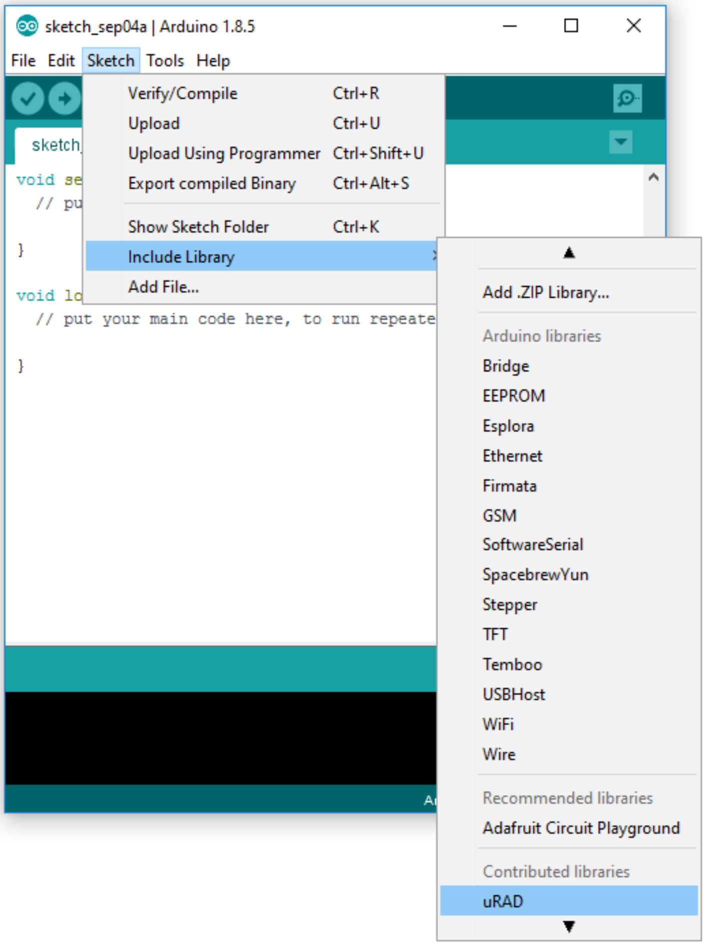
This manual is intended for those who already know the Arduino basics. If you are new in Arduino, we recommend visiting www.arduino.cc/en/Guide/HomePage to learn how to use your Arduino board and install Arduino IDE.

If you have the uRAD tutorial for Arduino, you will find complete information about getting started with Arduino from the beginning.

Library Installation

We have created for you several functions that will help you to setup and use uRAD in the easiest way. To be able to use them, first, you must install uRAD library for Arduino:

1. Download from www.urad.es/en/mi-cuenta/downloads the ***Arduino_software.zip*** folder and save it into your hard-disk. Unzip it and inside *Arduino_software\Library*, you will find the library ***uRAD.zip***.
2. Open the Arduino IDE and navigate to *Sketch > Include Library > Add. ZIP Library*. Navigate to the ***uRAD.zip*** location and open it.
3. Return to the *Sketch > Include Library* menu. You should now see the library at the bottom of the drop-down menu.



Once the library is installed, in your Sketch, you must create an object of your uRAD class. This allows you to use, under a specific name, the two available functions that controls your uRAD shield. First, write in your code, before the setup:

```
#include <uRAD.h>  // include the installed library
uRAD my_uRAD;      // create the object with the name my_urad
```

Now, you can use the functions `my_uRAD.loadConfiguration(…)`, and `my_uRAD.detection(…)`.

Load Configuration

uRAD set up is very straightforward. You must include one simple function in your Arduino code to select your configuration.

```
my_uRAD.loadConfiguration(mode, f0, BW, Ns, Ntar, Rmax, MTI, Mth)
```

As you already know, there are 8 configuration parameters, see Chapter 2. You must introduce all of them in this function to configure uRAD. All these parameters are integer numbers of type "*byte*" (*uint8_t*). The following table summarize the values that can be introduced.

`mode`	1	2	3	4
`f0`	5 to 245	5 to 195	5 to 195	5 to 195
`BW`	-	50 to (245 – f0)	50 to (245 – f0)	50 to (245 – f0)
`Ns`	50 to 200	50 to 200	50 to 200	50 to 200
`Ntar`	1 to 5	1 to 5	1 to 5	1 to 5
`Rmax`	0 to 75	1 to 100	1 to 100	1 to 100
`MTI`	-	0 or 1	0 or 1	0 or 1
`Mth`	1 to 4	1 to 4	1 to 4	1 to 4

⚠ *WARNING:* Entering a forbidden value in a parameter will result in its default value. Default values are: mode = 3, f0 = 5, BW = max available, Ns = 200, Ntar = 1, Rmax = 75 or 100, MTI = 0, Mth = 4.

- Include this function at least one time, always before your first target detection.
- The configuration does not change until you call again the function.
- Include this function, in any part of you code, as many times as you want to update the configuration for the next target detection.

Switch ON/OFF

Digital pin 6 switch ON/OFF uRAD.

- Write `digitalWrite(6, HIGH)` to switch ON uRAD.
- Write `digitalWrite(6, LOW)` to switch OFF uRAD.

You can set the configuration for the first time or update it with `my_uRAD.loadConfiguration(…)`, independent of whether uRAD is ON or OFF. It is recommended to add a short delay: `delay(100)` after powering ON uRAD.

Target Detection

Obtaining the detected information from uRAD is as simple as calling one function. Every time you want to obtain the info, you should include in your code:

```
my_uRAD.detection(distance, velocity, SNR, movement, I, Q)
```

This function returns to you 6 arrays with the detected information. Available returned information depends on the configuration mode.

	Mode	Array size	Type of variable
`distance`	2, 3, 4	Ntar	float
`velocity`	1, 3, 4	Ntar	float
`SNR`	All	Ntar	float
`movement`	All	1	boolean
`I`, `Q`	All	Ns (mode 1, 2) 2*Ns (mode 3) 3.5*Ns (mode 4)	uint16_t

When in mode 3, the triangular ramp has Ns ascending samples and Ns descending samples, 2*Ns in total. On the other hand, mode 4, dual rate, provides two triangular ramps with different duration. The first one has Ns samples ascending and descending, and the second one 0.75*Ns samples ascending and 0.75*Ns descending. In total 3.5*Ns samples.

WARNING: Declare always first all the arrays with their corresponding sizes and types, at the beginning of your code. It is only necessary to declare those arrays that you will ask for. Depending on your Arduino type, you will not be able to setup high Ns and ask for I, Q when in mode 4 due to RAM memory constrains.

- uRAD must always be switched ON before using `my_uRAD.detection(…)`.
- If the information is not available due to the mode selection or because there is not any detected target, `my_uRAD.detection(…)` returns 0 in the corresponding array.
- Target information is listed in each array position from higher to lower SNR.
- If you do NOT need any of the provided info, type 0 in the corresponding input of `detection` function.

For example, `my_uRAD.detection(distance, 0, 0, movement, 0, 0)` gives you only the distance to the detected targets and the alert/flag whether some target is in movement.

ADVICE: Do not ask for I and Q unless you are going to use them, because this involves a large amount of transmitted data from uRAD to Arduino and therefore, the update rate decreases as well as the free RAM memory.

Example. Movement Detector

One of the simplest applications is to use uRAD as a movement detector. In the following code, we print through the serial port a warning message when some movement is detected.

```
#include <uRAD.h>    // include the library

uRAD my_uRAD;        // create the object my_uRAD

// input parameters
uint8_t mode, f0, BW, Ns, Ntar, Rmax, MTI, Mth;   // variable definition

// results ouput array
bool movement[1];

void setup() {
  Serial.begin(250000);    //serial port baud rate

  mode = 2;   // sawtooth mode
  f0 = 5;     // starting at 24.005 GHz
  BW = 240;   // using all the BW available = 240 MHz
  Ns = 200;   // 200 samples
  Ntar = 1;   // only one target of interest
  Rmax = 25;  // searching in a range of 25 m
  MTI = 0;    // MTI mode disabled (irrelevant for this application)
  Mth = 4;    // most sensitive threshold

  // load the configuration
  my_uRAD.loadConfiguration(mode, f0, BW, Ns, Ntar, Rmax, MTI, Mth);

  digitalWrite(6, HIGH);   // switch ON uRAD
  delay(100);   // recommended delay after powering ON uRAD
}

void loop() {
  // target detection request
  my_uRAD.detection(0, 0, 0, 0, 0, movement);

  // print the results through the serial port
  if (movement[0] == true){
    Serial.print("Alert. Intruded detected.");
  }
}
```

ADIVCE: Test different values of `Mth` to see how the sensitivity of the alert changes.

This example can be found in the library *uRAD/examples/uRAD_movement_detector*

Further examples can be also found in the library *uRAD/examples*

Graphical User Interface 4

Learn to configure and use uRAD in the easiest way

The Graphical User Interface (GUI) is and executable software that allows you to configure and run uRAD, as well as visualize the reflected signal to understand where the results come from.

GUI Installation

The GUI is a full software developed in Python that does not need any installation to run. However, it interacts with the Arduino board and therefore, to be able to use the GUI, you must first install and additional library for Arduino:

1. Download from www.urad.es/en/mi-cuenta/downloads the ***Arduino_software.zip*** folder and save it into your hard-disk. Unzip it and inside *Arduino_software\Library*, you will find the library ***uRAD_GUI.zip***.
2. Open the Arduino IDE and navigate to *Sketch > Include Library > Add. ZIP Library*. Navigate to the ***uRAD_GUI.zip*** location and open it.
3. Return to the *Sketch > Include Library* menu. You should now see the library at the bottom of the drop-down menu.
4. Now, go to *File > Examples > uRAD_GUI > uRAD_load_GUI*. This opens the sketch ***uRAD_load_GUI.ino***. Upload it to your board. This sketch can also be found if you unzip ***uRAD_GUI.zip*** and go to *uRAD_GUI\examples\ uRAD_load_GUI*.

On **Windows**:

5. Navigate to *Arduino_software\GUI* and click with the right button on the file ***InstallFont.bat***. Select *Run as administrator*. This step installs a font to correctly visualize the GUI.
6. Once the ***uRAD_load_GUI.ino*** is uploaded in your board and the font installed, you just need to navigate to *Arduino_software\GUI* and execute the program ***uRAD_GUI***. This file is like a shortcut of ***uRAD.GUI.exe*** that is located in *Arduino_software\GUI\build\ exe.win32-3.6*.

On **Linux**:

5. You must firs install Python 3 to run GUI, whether you have not installed it yet. Open a terminal and type the following commands:

```
sudo apt-get install python3
sudo apt-get install python3-pip
```

6. You must also install the necessary Python libraries. Type in the terminal:

```
sudo pip3 install pyqt5
sudo pip3 install numpy
sudo pip3 install pyqtgraph
sudo pip3 install pyserial
sudo pip3 install datetime
```

7. To launch the GUI, execute in Python the program ***uRAD_GUI.py*** located in the folder *Arduino_software_linux\GUI* or type in the terminal:

```
python3 uRAD_GUI.py
```

On **Mac**:

5. You must firs install Python 3 to run GUI, whether you have not installed it yet. Open a terminal and type the following commands:

```
brew install python3
```

6. You must also install the necessary Python libraries. Type in the terminal:

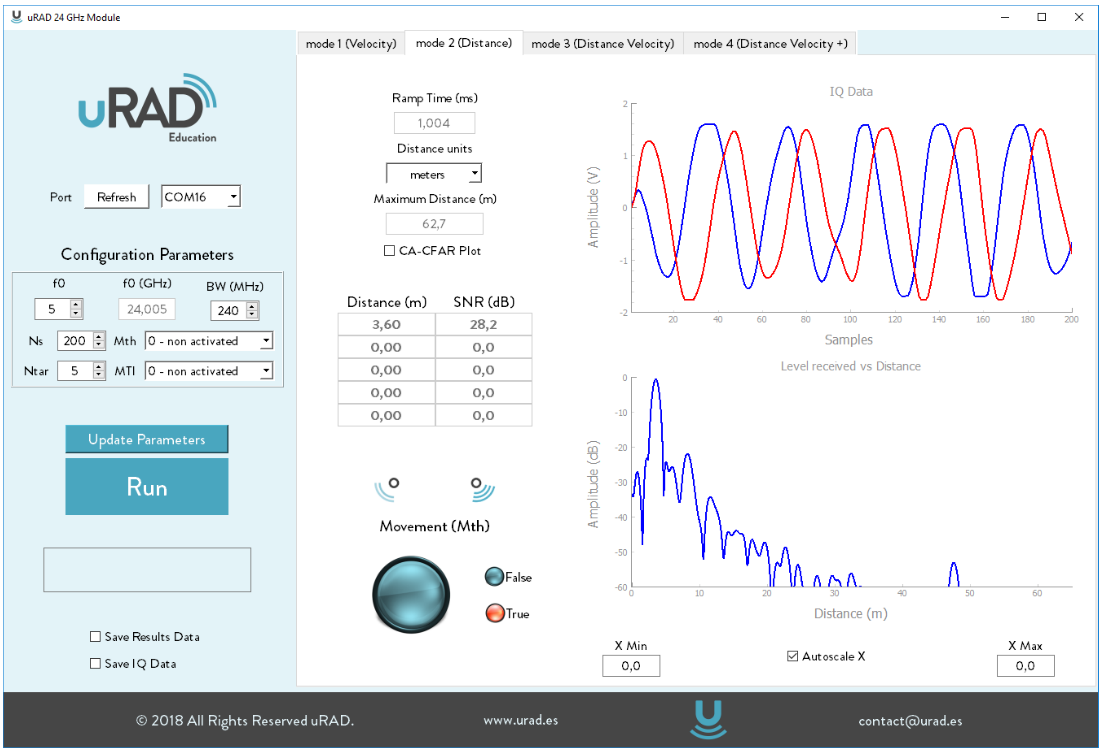
```
pip3 install pyqt5
pip3 install numpy
pip3 install pyqtgraph
pip3 install pyserial
pip3 install datetime
```

7. To launch the GUI, execute in Python the program ***uRAD_GUI.py*** located in the folder *Arduino_software_mac\GUI* or type in the terminal:

```
python3 uRAD_GUI.py
```

Using the GUI

After uploading ***uRAD_load_GUI.ino*** in your Arduino board and installing the font, navigate to *Arduino_software\GUI* and execute the program ***uRAD_GUI***. You will see a window similar to the following image:



uRAD 24 GHz Module

mode 1 (Velocity) | mode 2 (Distance) | mode 3 (Distance Velocity) | mode 4 (Distance Velocity +)

Port: COM16

Configuration Parameters

f0: 5
f0 (GHz): 24,005
BW (MHz): 240
Ns: 200
Mth: 0 - non activated
Ntar: 5
MTI: 0 - non activated

□ Save Results Data
□ Save IQ Data

Ramp Time (ms): 1,004
Distance units: meters
Maximum Distance (m): 62,7
□ CA-CFAR Plot

Distance (m)	SNR (dB)
3,60	28,2
0,00	0,0
0,00	0,0
0,00	0,0
0,00	0,0

X Min: 0,0
☑ Autoscale X
X Max: 0,0

© 2018 All Rights Reserved uRAD. www.urad.es contact@urad.es

The GUI is very intuitive and it gathers the parameter configuration and the output results in a visual way. Next, we briefly explain the different parts of the GUI.

The blue left column contains:

- **Port:** select the USB port where uRAD is connected.
- **Configuration Parameters:** here you introduce the parameters that configures uRAD. The mode is selected with the tabs of the upper part.
- **Update Parameters button:** each time you change one of the Configuration Parameters, click this button to update the new configuration.
- **Run/Stop button:** click to start or stop uRAD.
- **Text box:** displays a text when some error occurs.
- **Save Results Data checkbox:** when you check this checkbox, a **results.txt** file is created in the folder *GUI\OutputFiles* with the results information. When you uncheck the box, the program stops saving. This file is never overwritten, the program writes the results consecutively when you check and uncheck the box. Every line in the **results.txt** file is one measurement and it has 21 columns. Column 1 is the mode. Columns from 2 to 16 is distance, velocity and SNR of the 5 possible targets (0 indicates no result).

Column 17 is the Mth value and column 18 is 1 if movement = true or 0 if movement = false. Column 19 is the MTI value. Column 20 and 21 is the date and time.

- Save IQ Data checkbox: when you check this checkbox, several .txt files are created with the I and Q values. The files depend on the mode. In mode 1, two files: ***I_CW.txt*** and ***Q_CW.txt*** are created. In mode 2, two files: ***I_FMCW_sawtooth.txt*** and ***Q_FMCW_sawtooth.txt***. In mode 3, four files: ***I_up_FMCW_triangle.txt***, ***Q_up_FMCW_triangle.txt*** for the results of the up ramp, and ***I_down_FMCW_triangle.txt***, ***Q_down_FMCW_triangle.txt*** for the down ramp. In mode 4, eight files: ***I_up_1_FMCW_triangle_DualRate.txt***, ***Q_up_1_FMCW_triangle_DualRate.txt*** for the first up ramp; ***I_down_1_FMCW_triangle_DualRate.txt***, ***Q_down_1_FMCW_triangle_DualRate.txt***, for the first down ramp; ***I_up_2_FMCW_triangle_DualRate.txt***, ***Q_up_2_FMCW_triangle_ DualRate.txt*** for the second up ramp; ***I_down_2_FMCW_triangle_DualRate.txt***, ***Q_down_2_FMCW_triangle_DualRate.txt***, for the second down ramp. Each line in these files is one measurement. Each line has so many columns as number of samples (Ns) are selected plus two additional columns with the date and time. When you uncheck the box, the program stops saving. These files are never overwritten, the program writes the results consecutively when you check and uncheck the box.

The main white window shows the results and displays:

- Four tabs to select the mode: depend on the mode the displayed information can varied slightly.
- Distance and velocity units: select meters, feets or yards and m/s, km/h or mph.
- Maximum distance and velocity: max. distance and velocity according to BW and Ns.
- Ramp time: displays the actual duration of the ramps according to BW and Ns.
- Results table: displays the results of the detected targets.
- Movement (Mth) led: displays whether a movement is detected, if Mth is activated in Configuration Parameters.
- IQ Data plot: displays de amplitude of the reflected signal in volts: I (blue) and Q (red). In mode 3 and 4, solid lines are I,Q of up ramps and dashed lines of down ramps.
- Level received vs Distance/Velocity plot: displays the spectrum of the received signal, calculated by means of the Fast Fourier Transform (FFT) of the complex signal formed by I + *j*Q. The FFT allows us to identify the targets that corresponds with the peaks of the spectrum. The x-axis is Distance and the y-axis is the Amplitude normalized to 0 dB maximum. Each line in this plot correspond with the spectrum of each up and down ramp.
- Plot Peak Algorithm checkbox: checking this block displays a red line in the spectrum. Every peak that is under this line is identified as a target.
- Distance vs Velocity Display plot: displayed only in mode 4. It displays in a 2D plot the detected targets with a red dot, in function of its Velocity (y-axis) and Distance (x-axis). The size and transparency of the dot depend on the target SNR.

Safety & Handling 5

This chapter includes important safety and handling information for uRAD.

Read all safety and handling information below as well as the operating instructions before using uRAD in order to avoid any injury or damage.

Keep this user guide on hand for future reference.

Important Safety Information

⚠ *WARNING:* Failure to follow this safety instructions could result in fire, electric shock, or other injury or damage.

Proper handling uRAD contains sensitive electronic components. Do not drop, disassemble, crush, bend, deform, puncture, shred, microwave, incinerate, paint, or insert foreign objects into uRAD.

Water and wet locations Do not expose uRAD to water or rain, or handled near washbasins or other wet locations without a proper case. Take care not to spill any food or liquid on uRAD. In case uRAD gets wet, unplug from Arduino before cleaning, and allow it to dry thoroughly before turning it on again. Do not attempt to dry uRAD with an external heat source, such as a microwave oven or hair dryer.

uRAD repairs Never attempt to repair or modify uRAD by yourself. Disassembling may cause damage that is not covered under the warranty. If uRAD is damaged, malfunctions, or comes in contact with liquid, contact us at contact@urad.es.

Radio frequency interference Observe signs and notices that prohibit or restrict the use of radio frequency devices. Emissions from uRAD can negatively affect the operation of other radio frequency equipment operating in the same frequency band. Turn off uRAD when use is prohibited, such as traveling in aircraft, or when asked to do so by authorities.

Important Handling Information

⚠ *WARNING:* Failure to follow this handling instructions could result in damage to uRAD or other property.

Carrying uRAD contains sensitive electronic components. Do not bend, drop or crush it.

Cleaning To clean use a soft lint-free tip and isopropyl alcohol. Dust can be removed with compressed air of low power.

Plugging Never force the connector or apply excessive pressure because this may cause damage that is not covered under the warranty. Check for obstructions and make sure that uRAD connectors matches Arduino connectors. In some Arduinos, plugging can be harder.

Operating Temperature Keeping uRAD within acceptable temperatures. uRAD components operate from -40ºC to 85ºC but we recommend operates uRAD in the range from -20ºC to 65ºC.

Disposal and Recycling Information Your uRAD must be disposed of properly according to local laws and regulations. Because this product contains electric components, the product must be disposed of separately from household waste. Contact your local authorities to learn about recycling options.

Product Warranty 6

Manufacturing

All components and solder alloys used in this product comply with the RoHS Directive. The RoHS Directive prevents all new electrical and electronic equipment placed on the market in the European Economic Area from containing more than agreed levels of lead, cadmium, mercury, hexavalent chromium, poly-brominated biphenyls (PBB) and poly-brominated diphenyl ethers (PBDE).

Testing

Each uRAD shield is subject to strict tests to make sure they are not faulty:

- First, it is thoroughly tested for short circuits and open connections.
- Second, it is powered to check there are no over-range voltage.
- Then, the microcontroller is programmed and debugged.
- Finally, the board is plugged in an Arduino and several test programs are run to check its overall functionality.

Limited Warranty Statement

IMPORTANT: BY USING uRAD PRODUCTS YOU ARE AGREEING TO BE BOUNDED BY THE TERMS OF THIS LIMITED WARRANTY STATEMENT. DO NOT USE YOUR PRODUCTS UNTIL YOU HAVE READ THE TERMS OF THE WARRANTY. IF YOU DO NOT AGREE TO THE TERMS OF WARRANTY, DO NOT USE THE PRODUCTS AND RETURN THEM. THIS LIMITED WARRANTY IS THE END-USER'S SOLE AND EXCLUSIVE REMEDY AGAINST uRAD, WHERE PERMITTED BY LAW.

1. Warranties

1.1 uRAD warrants that its products will conform the specifications detailed in the corresponding datasheet. Warranty lasts for 1 year from the date of sale if the shield is bought outside the EU and last for 2 years if bought in the EU. uRAD shall not be liable for any defects that are caused by neglect, misuse or mistreatment, including any products that have been altered or modified by any way by the Customer.

1.2 If any uRAD product fails to conform to the warranty set forth above, uRAD's sole liability shall be to replace or repair such products. uRAD's liability shall be limited to products that are determined by uRAD not to conform to such warranty. If uRAD elects to replace or repair such products, uRAD shall be given a reasonable time to provide replacements. Replaced or repaired products shall be warranted for a new full warranty period.

1.3 The Customer agrees no to use uRAD products for any applications or in any components used in life support devices or to operate nuclear facilities or for use in other mission-critical applications or components where human life or property may be at stake. The Customer acknowledges and agrees that any such use is solely at the Customer's risk, and that the Customer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

1.4 uRAD may provide technical, applications or design advice. The Customer acknowledges and agrees that providing these services shall not expand or otherwise alter uRAD's warranties, as set forth above, and that no additional obligations or liabilities shall arise from uRAD providing such services.

1.5 uRAD disclaims all other warranties, expressed or implied, regarding products, including, but not limited to, any implied warranties of merchantability or fitness for a particular purpose.

1.6 The Customer acknowledges and agrees that the Customer is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning the products and any use of uRAD products in the Customer's applications, notwith- standing any applications-related information or support that may be provided by uRAD.

1.7 In no event shall uRAD be liable to the Customer or any third parties for any special, collateral, indirect, punitive, incidental, consequential or exemplary damages in connection with or arising out of the products provided hereunder, regardless of whether uRAD has been advised of the possibility of such damages. This section will survive the termination of the warranty period.