uRAD

User Manual

Arduino version

Software SDK v1.1

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. Furthermore, we have PCB interfaces to connect uRAD with Arduino MKR boards. Contact us for further information.

Release Information

The development software version SDK v1.1 only works for Arduino uRAD models with serial number higher than SN1942.

Additional Information

Hardware released:	uRAD v1.0	15/07/2018
	uRAD v1.1	15/05/2020
Software released:	SDK v1.0	15/07/2018
	SDK v1.1	15/02/2021

Purchase: www.uRAD.es/en/arduino

Software download: www.uRAD.es/en/mi-cuenta/downloads (only available with purchase)

Contact: contact@uRAD.es

uRAD Basics

2

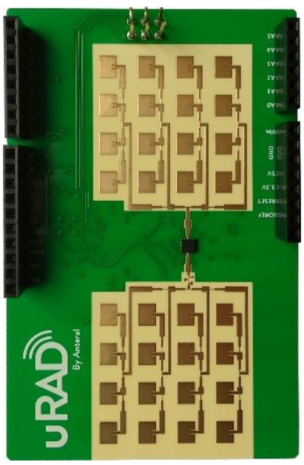
Read this chapter to learn about the features of uRAD, how to use it, and more.

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

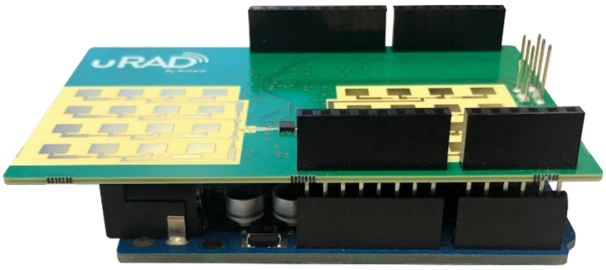
Bottom view

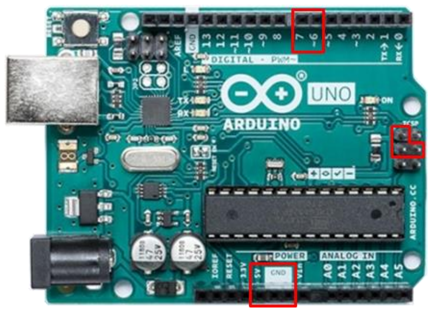
Side view

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:* Depend on the Arduino model, SPI lines are also internally connected to some digital pins besides ICSP connector. For example, in Arduino UNO, at pins 10, 11, 12, 13. We recommend do not use that shared pins because during data transmission those pins are used for SPI communication.

> *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. Most thin plastics are practically invisible to emission.

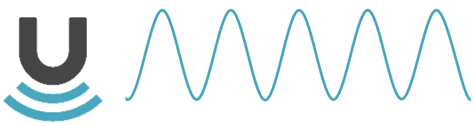
Main Features

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.

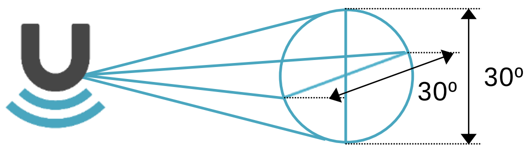
The main features of uRAD are:

Frequency Bandwidth

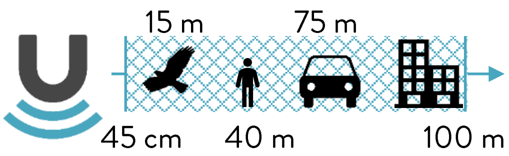


24.005 – 24.245 GHz

Field of View

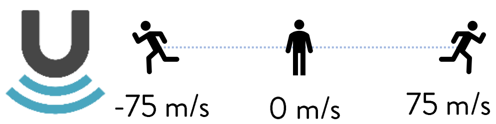


Distance Range



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.

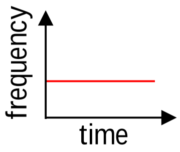
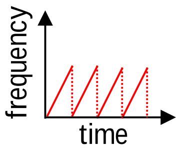
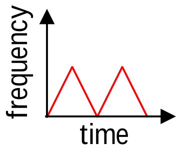
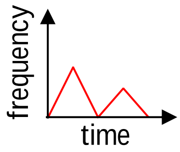
Additional information regarding technical features can be found in www.urad.es/en.

Configuration Parameters

You must enter 9 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)	0.45 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)	±0.7 to ±75	-	±0.2 to ±75	±0.2 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)	68	45	28	16

- [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]	68 to 43	45 to 38	28 to 24	16 to 15

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 set 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 the information of 3 targets (if there are) that reflect more power and 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. Therefore, you can detect targets farther than 100 meters.

7. MTI

It activates or deactivates 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.

9. Alpha

Searching targets consists in looking for peaks in the spectrum of the total received signal. This peak detection is done in the firmware of uRAD by means of a CA-CFAR algorithm (cell average – constant false alarm rate). Alpha is a parameter of this algorithm that allows to be more or less restrictive in the peak detection. The amplitude of each peak has to be "alpha dBs" greater than its surrounding zone of the spectrum to be considered as a detected peak.

Values of alpha too high results in a low probability of detection. On the contrary. Values too low results in a high probability of detection but false alarms rise. For each scenario, there would be a best value of alpha. Minimum selectable value of alpha is 3 and maximum value is 25.

alpha	Mode 1, 2, 3, 4
Range value	3 (high) to 25 (low) sensitivity

ADVICE: In mode 1, we recommend to start trying with a value of alpha = 20. In mode 2,3,4 we recommend a value of alpha = 10. Try several values of alpha to find the value that better fits your particular scenario.

Detected Information

uRAD provides you with complete information of its detection range:

- NtarDetected: returns the number of detected targets being the maximum number defined by the configuration parameter Ntar.
- 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
NtarDetected	All	-
Distance	2, 3, 4	meters
Velocity	1, 3, 4	meters/seconds
SNR	All	dB
I, Q	All	Arbitrary units from 0 to 4095
Movement	All	TRUE/FALSE

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.

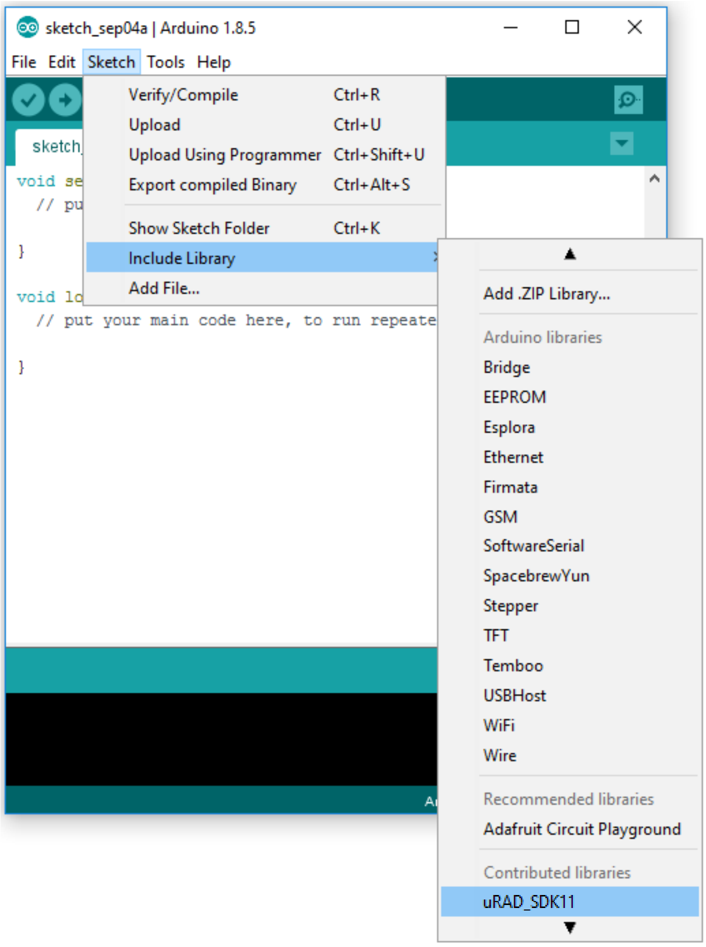
You can download the Arduino software from www.arduino.cc/en/Main/Software for Windows, Mac or Linux.

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.

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 *uRAD_Arduino_SDK11.zip* folder and save it into your hard-disk. Unzip it and inside *uRAD_Arduino_SDK11\Library*, you will find the library *uRAD_SDK11.zip* (SDK only available with purchase).
2. Open the Arduino IDE and navigate to *Sketch* > *Include Library* > *Add. ZIP Library*. Navigate to the *uRAD_SDK11.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_SDK11.h>  // include the installed library
uRAD_SDK11    uRAD;      // create the object with the name uRAD
```

Now, you can use the functions `uRAD.turnON(…)`, `uRAD.turnOFF(…)`, `uRAD.loadConfiguration(…)` and `uRAD.detection(…)`.

Switch ON/OFF

To switch ON/OFF uRAD, two functions have been programmed:

- `return_code = uRAD.turnON()`.
- `Return_code = uRAD.turnOFF()`.

These functions properly turn on/off the RF signal as well as set the digital pin number 6 High or Low level to enable or disable the power to the radar.

If `return_code = 0`, everything has run correctly inside the function, if `= -1`, some error has occurred.

Load Configuration

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

```
return_code = uRAD.loadConfiguration(mode, f0, BW, Ns, Ntar, Rmax, MTI, Mth, alpha, distance_true, velocity_true, SNR_true, I_true, Q_true, movement_true)
```

As you already know, there are 9 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
alpha	3 to 25	3 to 25	3 to 25	3 to 25

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, alpha = 3 or 25.

Finally, you have to introduce some variables to indicate which results and which not you would like to get: distance, velocity, SNR, raw data I and/or Q, and movement. These variables have to be declared of type *bool*. For example:

```
distance_true = true
velocity_true = true
SNR_true = true
I_true = false
Q_true = false
movement_true = false
```

If `return_code = 0`, everything has run correctly inside the function, if `= -1`, some error has occurred. Moreover, take into account the following points:

- Include this function at least one time, always before your first target detection.
- The configuration does not change until you call again the function.
- After turning off and turning on uRAD, do not forget to call again loadConfiguration to charge again your desired configuration.
- Include this function, in any part of you code, as many times as you want to update the configuration for the next target detection.

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:

```
    return_code = uRAD.detection(NtarDetected, distance, velocity, SNR, I, Q, movement)
```

This function returns to you 7 arrays with the detected information. Available returned information depends on the configuration mode.

	Mode	Array size	Type of variable
`NtarDetected`	All	1	uint8_t
`distance`	2, 3, 4	Ntar	float
`velocity`	1, 3, 4	Ntar	float
`SNR`	All	Ntar	float
`I`, `Q`	All	Ns (mode 1, 2) 2*Ns (mode 3) 2*Ns+2*ceil(0.75*Ns) (mode 4)	uint16_t
`movement`	All	1	

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. Ceil means rounded the number to the upper next integer number.

> *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 turned ON before using `uRAD.detection(…)`.
- If the information is not available due to the mode selection or because there is not any detected target, `uRAD.detection(…)` returns 0 in the corresponding array.
- Target information is listed in each array position from higher to lower SNR.
- If you have not declared some input variable because you do NOT need it, type 0 in the corresponding input of `detection` function.

> *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. Distance Meter

One of the simplest applications is to use uRAD as a distance meter. To open the example, go to *File* > *Examples* > *uRAD_SDK11* > *uRAD_SDK11_distance_meter.*

First, we include the uRAD library and define the object.

```
#include <uRAD_SDK11.h>// include the library

uRAD_SDK11 uRAD;       // create the object uRAD
```

Next, we define the input variables needed for uRAD as well as the output variables, those that we want uRAD to measure. In this case, they will be the *distance* and *SNR*.

```
// input parameters
uint8_t mode, f0, BW, Ns, Ntar, Rmax, MTI, Mth, alpha;
bool distance_true, velocity_true, SNR_true, I_true, Q_true, movement_true;
int return_code;

// results output array
uint8_t NtarDetected[1];
float distance[5], SNR[5];
```

Distance variable will be defined as *float*. This type can be as high as 3.4028235 E+38 and as low as -3.4028235 E+38 and it is saved as 32 bits (4 bytes) of information.

We define the baud rate of the serial port and below the value of each one of the input variables of the uRAD configuration function.

```
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 = 3;       // 3 target of interest
 Rmax = 100;     // searching along the full distance range
 MTI = 0;        // MTI mode disable because we want information of static
and moving targets
 Mth = 0;        // parameter not used because "movement" is not requested
 alpha = 10;     // signal has to be 10 dB higher than its surrounding
 distance_true = true;       // request distance information
 velocity_true = false;      // mode 2 does not provide velocity
information
 SNR_true = true;      // Signal-to-Noise-Ratio information requested
 I_true = false;       // In-Phase Component (RAW data) not requested
 Q_true = false;       // Quadrature Component (RAW data) not requested
 movement_true = false;      // not interested in boolean movement
detection
```

Now, we turn on uRAD and load the desired configuration.

```
// switch ON uRAD
   return_code = uRAD.turnON();
   if (return_code != 0) {
         Serial.println("Error turning on uRAD");
   }

   // load the configuration
   return_code = uRAD.loadConfiguration(mode, f0, BW, Ns, Ntar, Rmax, MTI,
Mth, alpha, distance_true, velocity_true, SNR_true, I_true, Q_true,
movement_true);
   if (return_code != 0) {
         Serial.println("Error configuring uRAD");
   }
```

Next, detection function is called.

```
void loop() {
  // target detection request
  return_code = uRAD.detection(NtarDetected, distance, 0, SNR, 0, 0, 0);
```

Finally, we print on screen through the serial port the distance and SNR of the different detected targets. To do this, we will formulate the condition that the SNR of those targets must be greater than 0 dB.

```
// iterate through desired targets
   if (return_code == 0) {
       for (uint8_t i = 0; i < NtarDetected[0]; i++) {
         // if target is reflective enough, print its distance
       if (SNR[i] > 0){
            Serial.print("Target: ");
            Serial.print(i+1);
            Serial.print(", Distance: ");
            Serial.print(distance[i]);
            Serial.print(" m, SNR: ");
            Serial.print(SNR[i]);
            Serial.println(" dB");
         }
       }
       // If number of detected targets is greater than 0 prints an empty
      line for a smarter output
       if (NtarDetected[0] > 0) {
         Serial.println(" ");
       }
   } else {
         Serial.println("Error in detection");
   }
```

Further examples can be found in *uRAD_Arduino_SDK11\Library\uRAD_SDK11\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 *uRAD_Arduino_SDK11.zip* folder and save it into your hard-disk. Unzip it and inside *uRAD_Arduino_SDK11\Library*, you will find the library *uRAD_GUI_SDK11.zip*.
2. Open the Arduino IDE and navigate to *Sketch* > *Include Library* > *Add. ZIP Library*. Navigate to the *uRAD_GUI_SDK11.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_SDK11* > *uRAD_load_GUI_SDK11*. This opens the sketch *uRAD_load_GUI_SDK11.ino*. Upload it to your board. This sketch can also be found if you unzip *uRAD_GUI_SDK11.zip* and go to *uRAD_GUI_SDK11\examples*.

On Windows:

5. Once the *uRAD_load_GUI_SDK11.ino* is uploaded in your board, you just need to navigate to *uRAD_Arduino_SDK11\GUI\GUI_windows* and execute the program *uRAD_GUI_SDK11*. This file is like a shortcut of *uRAD_GUI_SDK11.exe* that is located in *uRAD_Arduino_SDK11\GUI\GUI_windows\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_SDK11.py* located in the folder *uRAD_Arduino_SDK11\GUI\GUI_linux* 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:

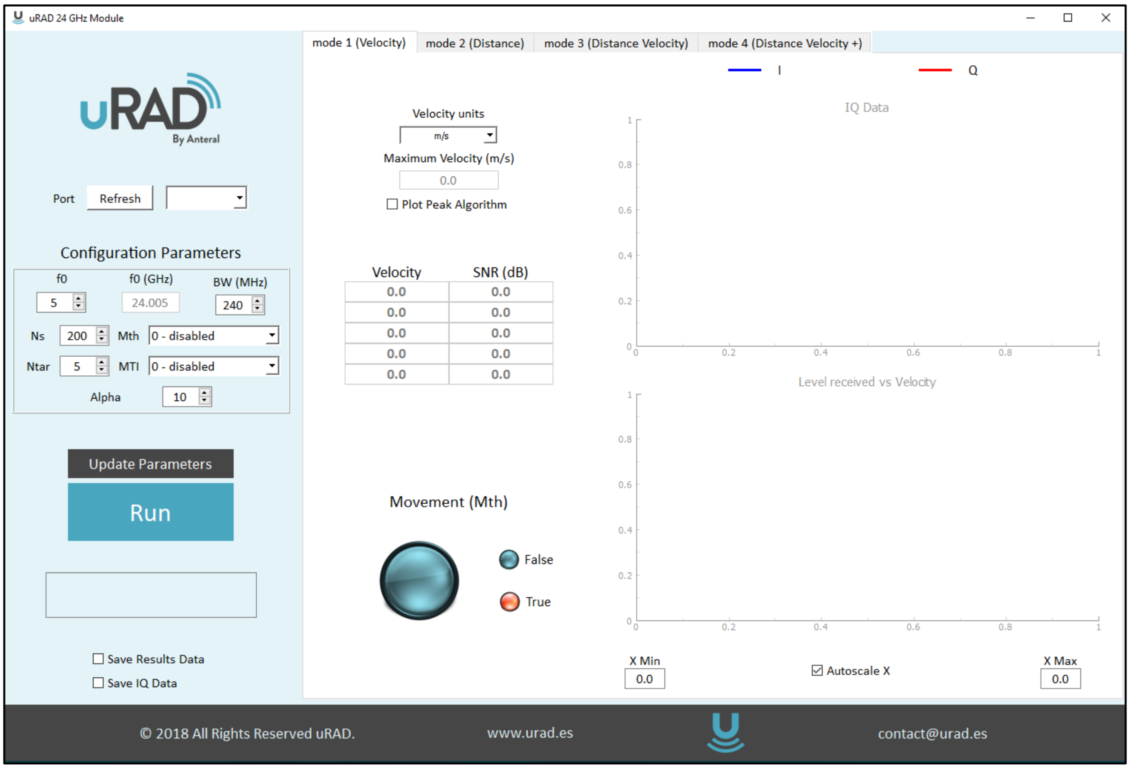
```
pip3 install pyqt5
pip3 install numpy
pip3 install pyqtgraph
pip3 install pyserial
pip3 install datetime
```

7. To launch the GUI, execute in Python the program *uRAD_GUI_SDK11.py* located in the folder *uRAD_Arduino_SDK11\GUI\GUI_mac* or type in the terminal:

```
python3 uRAD_GUI.py
```

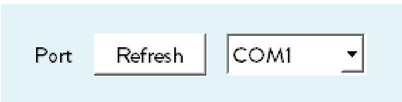
Selecting the Configuration Parameters

To launch the GUI, run the program *uRAD_GUI_SDK11*. After a few seconds, a window similar to the following will open.



The GUI is divided mainly into two parts. The left part on a light blue background, corresponds to the configuration parameters, power buttons and saving the results. The largest part on the right contains the visualization of the results. We will now explain the options present in the configuration parameters part.

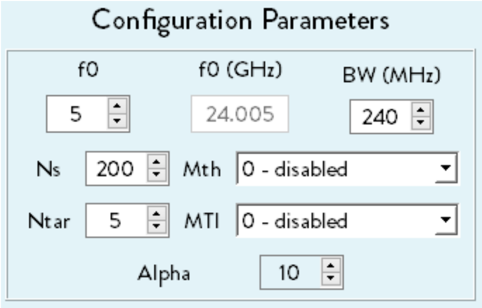
- USB Port: Below the uRAD symbol, there is the selection part of the USB port.



In the drop-down menu, you should select the USB port where your Arduino card is connected. If you have connected your card after launching the GUI, click *Refresh* to refresh the ports and see the USB port where you have it connected.

ADVICE: You can know the USB port where your Arduino is connected, going to the Windows Device Manager.

- *Configuration Parameters*: The following image shows the part to enter the configuration parameters.

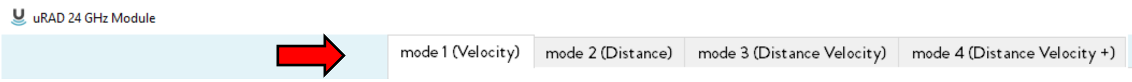


How you have already learned from Chapter 6 or from the User Manual, there 9 parameters that configure uRAD. In this box you can select 6 of them:

- f0: transmission frequency or starting ramp frequency (next to it is the real frequency value in GHz).
- BW: bandwidth in MHz (only selectable in modes 2, 3 and 4).
- Ns: number of samples.
- Ntar: number of targets.
- Mth: movement detection threshold.
- MTI: activating the mode Moving Target Indicator (discard static targets).
- Alpha: detection peak algorithm threshold.

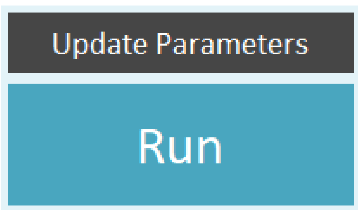
ADVICE: If you place the mouse a few seconds on the value of f0, BW, Ns, Ntar or Alpha without clicking, a small help will appear showing the range of values you can select.

For the selection of the operating mode, you must click on the corresponding tab at the top of the results window.



On the other hand, the parameter corresponding to Rmax/Vmax is not available as such in the GUI. If you want to visualize the results up to a certain distance or speed, it can be done by adjusting the axes of the graphs, as we will see later.

- *Update Parameters* and *Run/Stop button*: These two buttons are used to turn on and off uRAD and update the configuration.

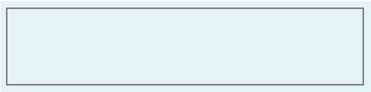


Click on the *Run* button to turn on uRAD. Once running, click on *Stop* to turn off uRAD.

If you change some value in *Configuration Parameters*, the change will not be effective until you click on the *Update Parameters* button.

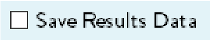
It is not necessary to stop uRAD to update the configuration, nor is it necessary to stop it to change the mode.

- Error text box: Below the *Run* button, there is a small empty box.



If an error occurs during the execution of the GUI, an error message will appear in this box.

- *Save Results Data* checkbox: In the lower part, there is a check box to save the results.



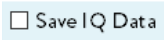
When you check this checkbox, the GUI starts saving automatically the distance, velocity and SNR results of the detected targets. A text file named *results.txt* is created in the folder *uRAD_Arduino_SDK11\GUI\...\OutputFiles.* When you uncheck the box, the program stops saving. This file is never overwritten, the program writes the results consecutively when you check and uncheck the box.

Every line in *results.txt* corresponds with one measurement and every measurement saves the information in up to 21 columns (depends on the detected targets):

- Column 1 is the mode.
- Each detected target creates 3 columns with its distance, velocity and SNR.
- Next column is the Mth value.
- Next column is 1 if movement = true or 0 if movement = false.
- Next column is the MTI value: 0 disable, 1 enable.
- Last two columns are date and time.

REMINDER: Depending on the mode, you will get results only of velocity and SNR (mode 1), only distance and SNR (mode 2), or velocity, distance and SNR (mode 3 and 4). If there is no result, the column will have a value equal to 0.

- *Save IQ data* checkbox: In the lower part, a box appears to save the RAW values of the I and Q components.



When you check this box, several .txt files are created with the values of I and Q. The files that are created depend on the mode. The following table will serve as a guide to know what files are created in each mode.

Modo 1	
I_CW.txt	I signal values with Ns samples
Q_CW.txt	Q signal values with Ns samples

Modo 2	
I_FMCW_sawtooth.txt	I ramp values
Q_FMCW_sawtooth.txt	Q ramp values

Modo 3	
I_up_FMCW_triangle.txt	I values of the ascending ramp
Q_up_FMCW_triangle.txt	Q values of the ascending ramp
I_down_FMCW_triangle.txt	I values of the descending ramp
Q_down_FMCW_triangle.txt	Q values of the descending ramp

Modo 4	
I_up_1_FMCW_triangle.txt	I values of the first ascending ramp
Q_up_1_FMCW_triangle.txt	Q values of the first ascending ramp
I_down_1_FMCW_triangle.txt	I values of the first descending ramp
Q_down_1_FMCW_triangle.txt	Q values of the first descending ramp
I_up_2_FMCW_triangle.txt	I values of the second ascending ramp
Q_up_2_FMCW_triangle.txt	Q values of the second ascending ramp
I_down_2_FMCW_triangle.txt	I values of the second descending ramp
Q_down_2_FMCW_triangle.txt	Q values of the second descending 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.

The exception is given in mode 4 in the files of the second ascending and descending ramp. Each measurement of those files has 0.75*Ns columns since that second ramp is shorter (plus two columns with the date and time).

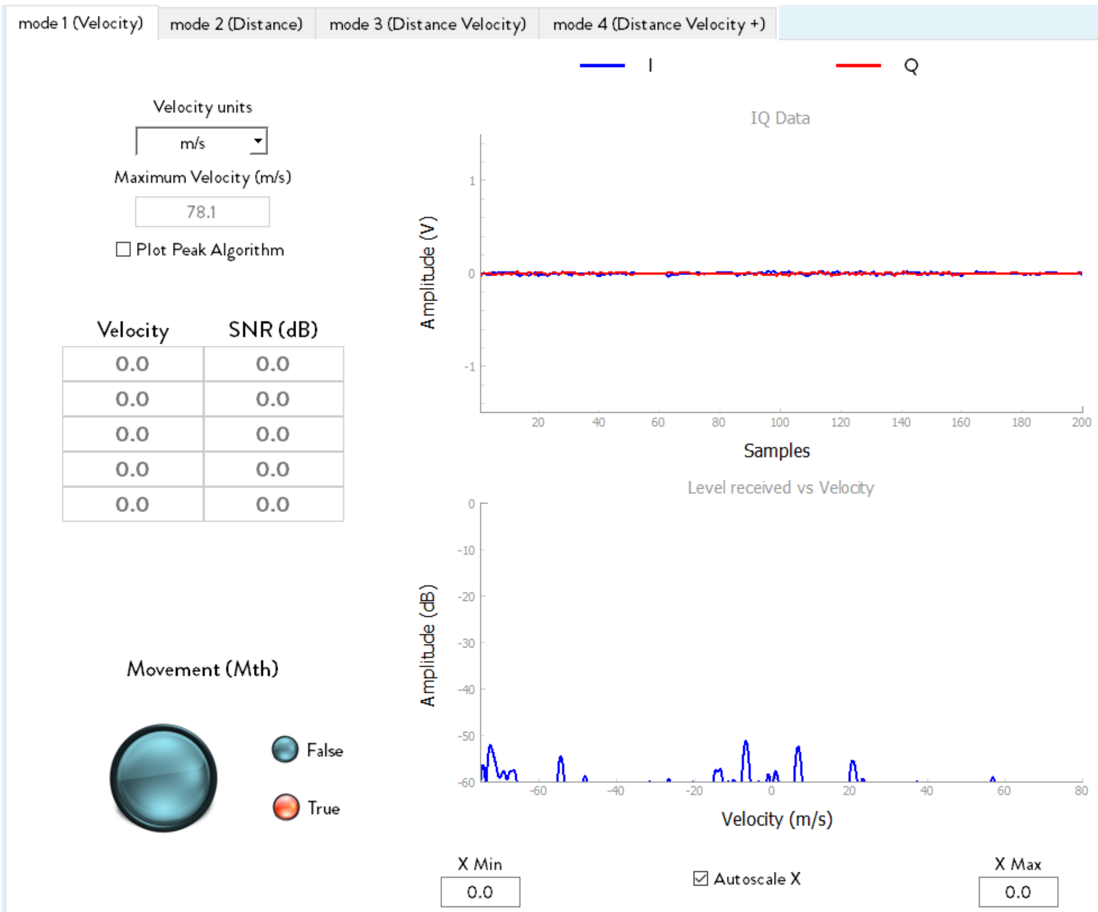
REMINDER: Depending on the mode, uRAD emits different signal with different ramps and therefore, the total received signal I and Q has different total length. For instance, with Ns = 100:

- if mode = 1 = 2 → length of I/Q = 100
- if mode = 3 → length of I/Q = 200 (100 + 100)
- if mode = 4 → length of I/Q = 350 (100 + 100 + 75 + 75)

When you uncheck the box, the program stops saving. These files are never overwritten, the program writes the results consecutively with their corresponding date and time.

Visualizing Results – Mode 1

Depending on the mode, the part of display the results changes slightly. In mode 1, the Doppler mode of only velocity, the appearance of the screen with the radar on is as follows:

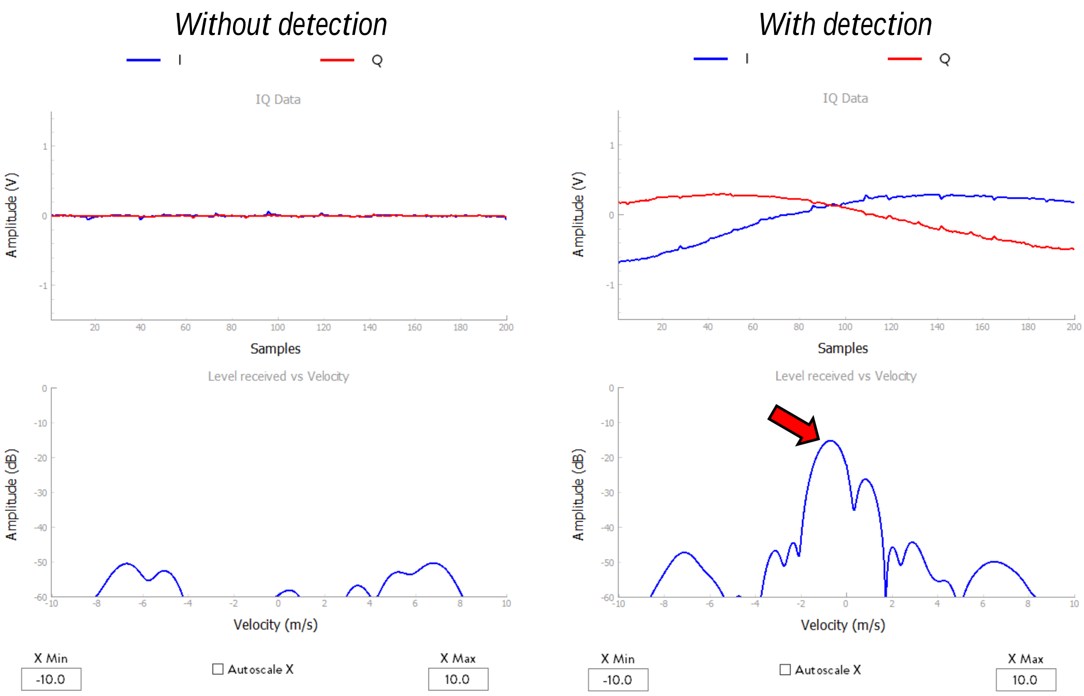


The upper right graph, titled *IQ Data*, show the I (blue) and Q (red) received signals. The x-axis corresponds with the number of samples, so whether you change *Ns* and click on *Update Parameters*, the graph axis changes. The signal amplitude of I and Q is shown in volts, represented in the y-axis.

The I and Q signals are the phase and quadrature components of the received signal by uRAD. In other words, by building the complex signal I + *j*Q you would get the total signal received and from there, you can extract the magnitude and phase of the reflected wave.

The lower right graph, titled *Level received Vs Velocity*, corresponds to the fast Fourier transform (FFT) of the complex signal I + *j*Q. The FFT is of great interest because from it, you can obtain the information of the targets. The FFT, also called signal spectrum, shows the frequency components that comprise the total signal. Then, each of the relevant peaks that appear in the spectrum corresponds with a frequency, and therefore with a target detected by uRAD. By observing the position of that peak with respect to the x-axis, the velocity of the objective can be calculated. The FFT is represented in dB, therefore, the y-axis is the amplitude of the signal in dB, which is normalized to a maximum of 0 dB and a minimum of -60 dB.

We see below two situations where nothing is detected and where a target is detected at a certain speed, to see the differences in the graphs.



It can be clearly seen that when there is no detected target, no signal appears in *IQ data* or peak in the spectrum. When a target is detected, the variation of the I and Q signals is seen and a peak appears in the spectrum.

WARNING: On the spectrum, actually, many peaks are present. Some are targets, others correspond to noise and others may be the result of interference, bounces, etc. A peak detection algorithm has been programmed to identify those that really correspond to objectives.

To the left of the graphs, in the central part, a table appears where the results are shown. These results are extracted from the previous graphs.

Velocity	SNR (dB)
-0.7	36.8
0.0	0.0
0.0	0.0

This table has as many rows as the number of targets, Ntar, you have selected. It shows the speed of the detected targets and their SNR in dB.

REMINDER: the SNR is the Signal to Noise Ratio. It is the difference in magnitude between the reflected signal due to the target and the noise background due to the whole system. It gives an idea of the amount of reflected signal from each target and, therefore, its size and reflectivity.

Negative velocities mean that the target approaches uRAD while positive velocities mean that the target is moving away from uRAD.

In the left-hand part above, you can select with the drop-down in which units the velocity results are displayed, *Velocity units*. You can select meters per second (m/s), kilometers per hour (km/h) and miles per hour (mph).

Velocity units

m/s

Below, an information box allows you to see the maximum velocity that can be measured. This maximum velocity depends only on the f0 in GHz selected:

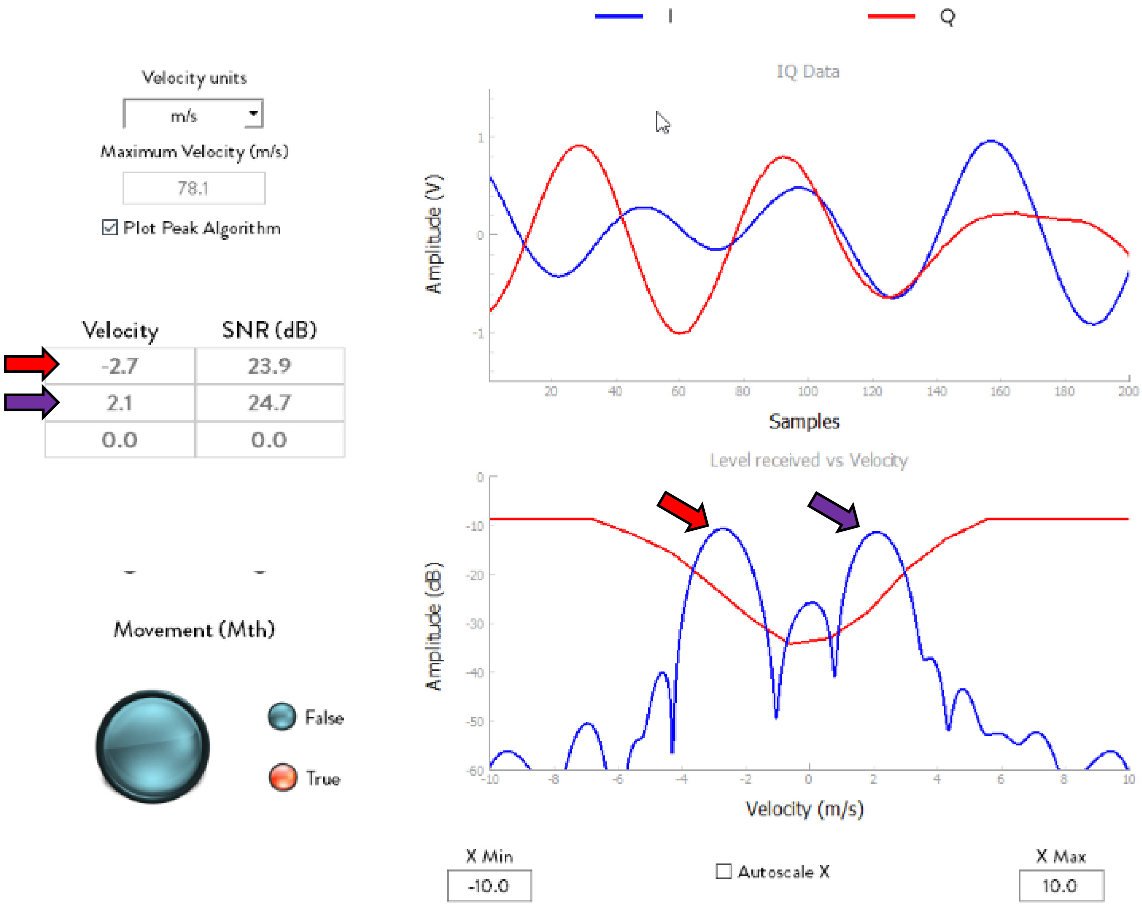
Maximum Velocity (m/s)

78.1

$$Velocity_{max} = \frac{1875}{f0\ (GHz)}$$

Just below, there is a checkbox named *Plot Peak Algorithm*. When this checkbox is activated, a red line appears on the spectrum graph. This line represents the peak detection algorithm that we mentioned above. In order for a peak to be considered a detected target, it must at least exceed this red line, although it must also comply with some other condition that prevents it from being discarded. Change the value of Alpha, click on *Update Parameters*, and you will see how the red line changes the level. If Alpha is smaller, more peaks will be detected.

See below an example where two objectives are detected, one approaching and another moving away, its relationship of peaks in the spectrum and the results in the table.



Finally, at the bottom left, there is a led indicating if some movement has been detected.

For this led to work, you must first activate it in *Configuration Parameters*, in the drop-down of the *Mth* parameter. The led will turn red if there is a movement detected and blue if there is not.

Visualizing Results – Mode 2

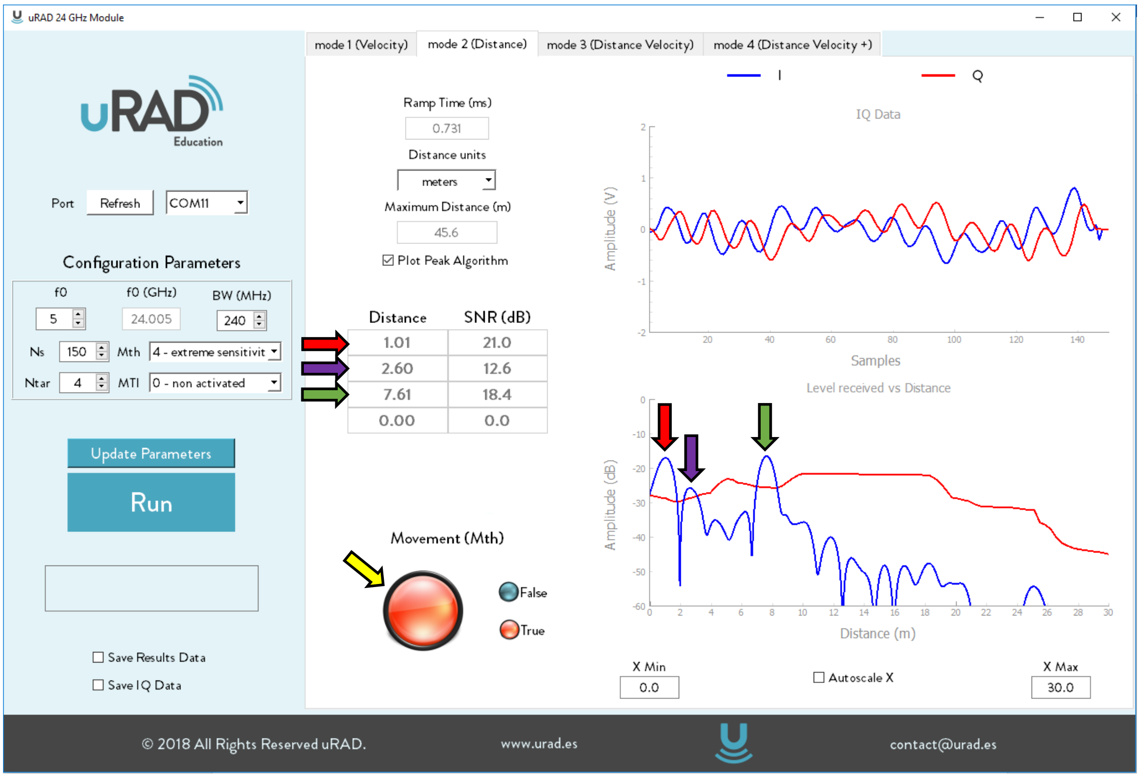
The results shown in mode 2 are similar to those in mode 1, except that in this mode only distance results are shown, instead of velocity.

The spectrum graph is titled in this case *Level received Vs Distance*. Because in this case the signal emitted is modulated in frequency by means of a ramp that varies in time, the x-axis of the spectrum corresponds to the distance of the targets to uRAD.

In the upper left part, the information of the ramp duration in milliseconds is shown, *Ramp Time* and the maximum detected distance, *Maximum Distance*. You can also select the *Distance units*. The maximum distance is a function of the configuration parameters according to the following formula:

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

We see below an example where 3 targets have been detected at 3 different distances and, in addition, one or several of them are in motion, since the movement LED is activated.

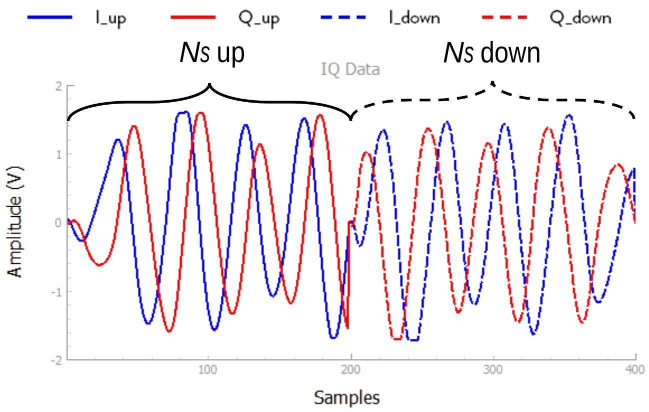


Visualizing Results – Mode 3

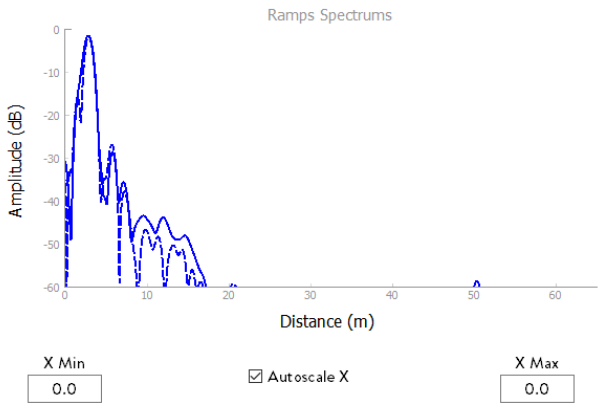
Mode 3 allows you to calculate both the velocity and distance of the targets because the signal emitted varies in frequency according to a triangular shape, with a ramp up and down.

> *ADVICE:* If you want to learn a little more about how different frequency ramps allow different results, we recommend reading the uRAD White Paper, available on www.urad.es.

The *IQ Data* graph, shows in this case the I and Q values for the up and down ramp. The signal received during the up ramp is drawn with a continuous line and the one received during the down ramp with dashed line. Therefore, the x-axis has twice as many samples as the selected Ns, since they are Ns for the up and Ns for the down. Let us see an example with Ns = 200, the maximum number of samples.

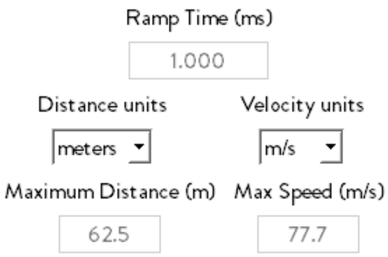


In the spectrum graph, two signals also appear. In the same way, the continuous line corresponds to the FFT of the complex signal I + *j*Q of the up ramp, and the discontinuous one with the FFT of the signal of the down ramp.



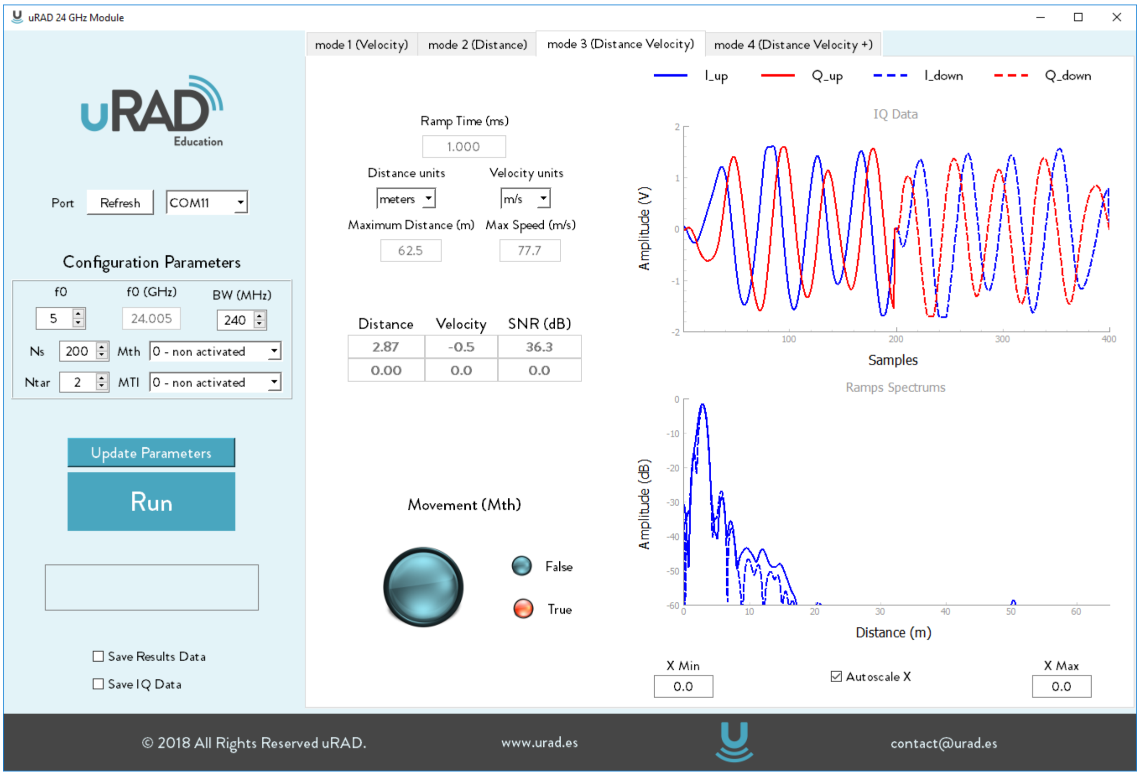
In this mode there is no option to paint the peak detection algorithm since we have considered that painting two more lines on this graph would be unclear.

In this mode, you can also select the units of both measured distance and velocity, as well as visualize information of ramp time, maximum velocity and distance. Therefore, the results table shows the distance, velocity and SNR information of the detected targets.



Distance	Velocity	SNR (dB)
2.87	-0.5	36.3
0.00	0.0	0.0

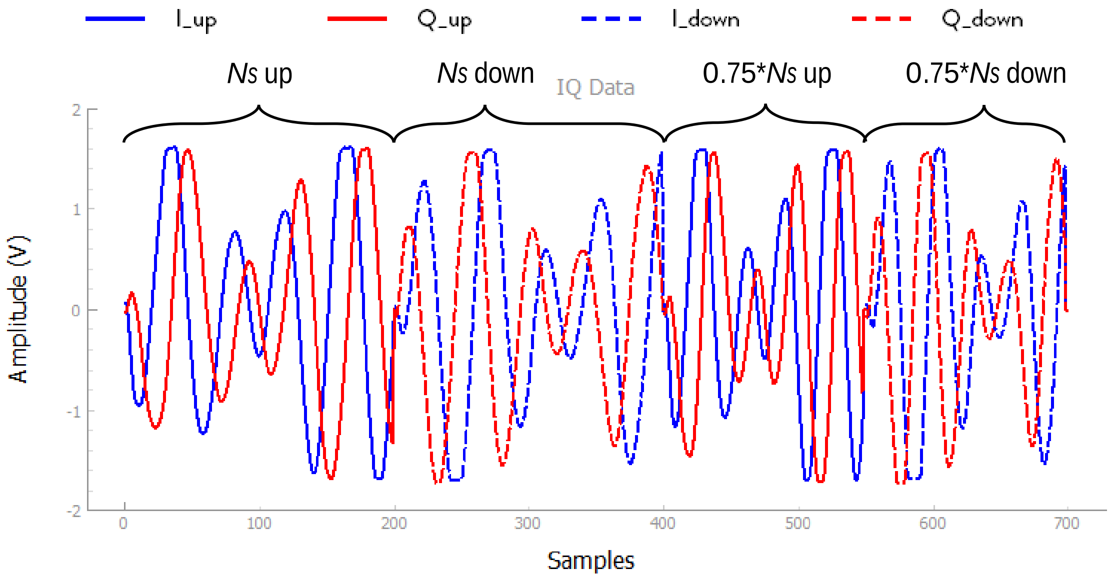
We see below a typical example, where a single target has been detected at 2.87 meters and approaching at velocity of 0.5 m/s.



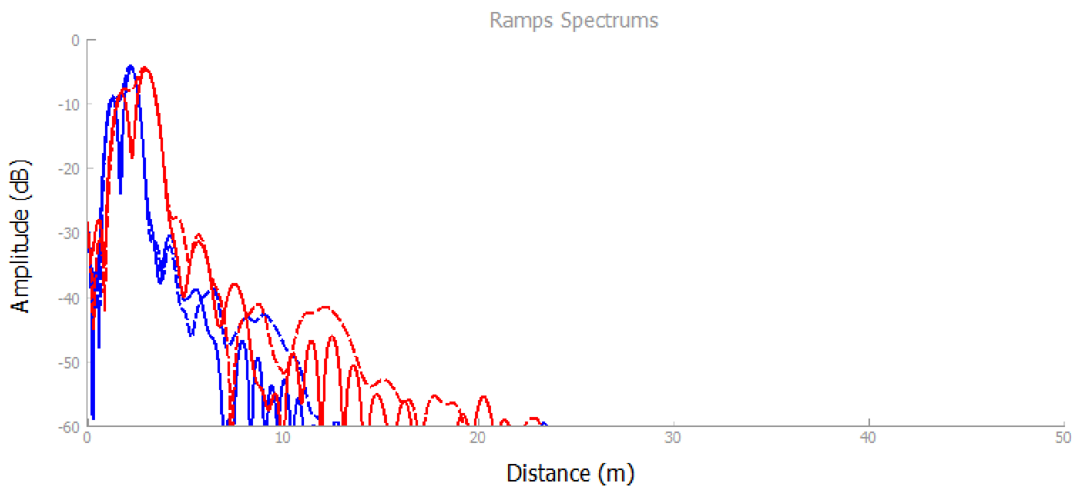
Visualizing Resuls – Mode 4

Mode 4 is very similar to mode 3, since it also allows obtaining velocity and distance information. In this mode, the transmitted signal also varies in frequency according to two consecutive triangular signals of different duration. The first triangular is formed by an up ramp and another down ramp, of Ns samples each one; and the second triangular by an up and down ramp of 0.75 * Ns samples each one.

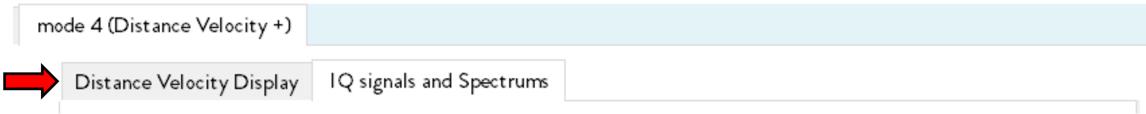
This can be easily seen in the *IQ Data* graphic. Let us see an example of a signal received with Ns = 200 samples.



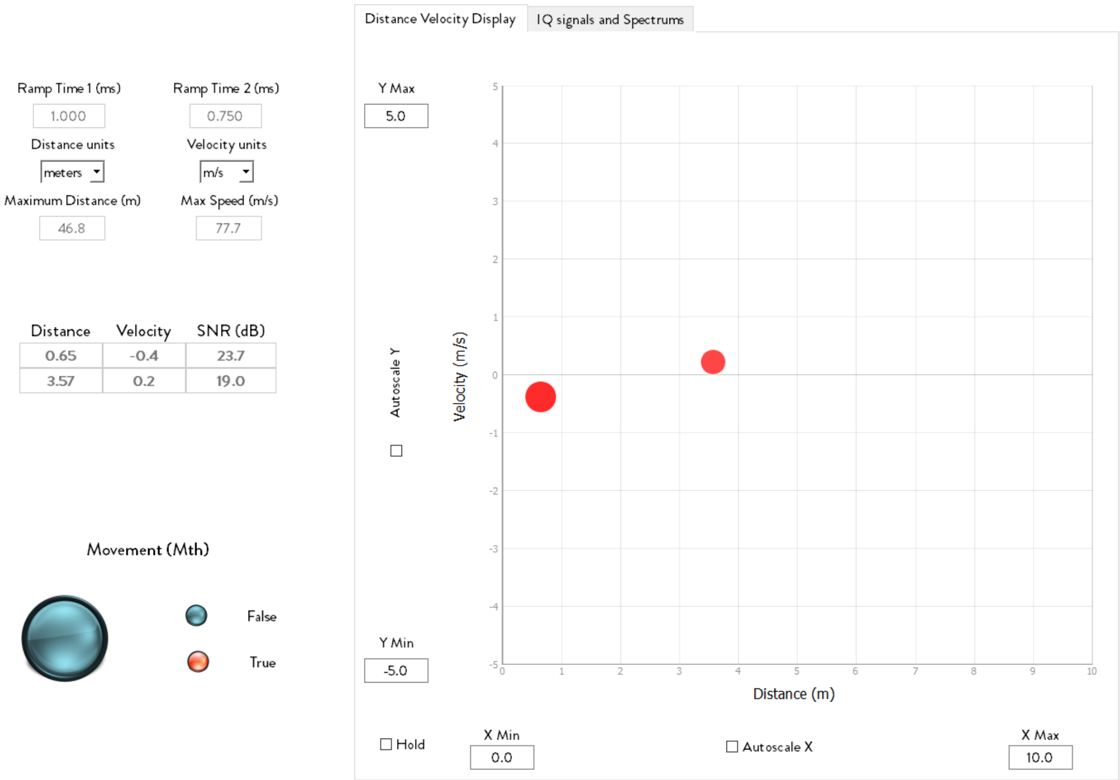
Consequently, the spectrum will also show 4 different signals. The blue continuous line corresponds with the FFT of the first up ramp, the blue discontinuous one, with the first down ramp, the red continuous one, with the second up ramp and the red discontinuous one with the second down ramp.



In mode 4, we have created an additional window to visualize the distance and velocity results of the targets in a graphical form. You can select it in the tab *Distance Velocity Display*.



Clicking on it, you will see a screen similar to this one. Each detected target is represented by a red dot. The x-axis corresponds to the distance at which the objective is located and the y-axis represents the velocity. The size and opacity of the objective is directly proportional to its SNR. Targets with high SNR will look bigger and less transparent.



Down to the left, there is a checkbox with the word *hold*. When activated, the results painted in this graph do not disappear until after a few seconds, being easier to visualize the path of the objectives.

The rest of options and information of mode 4 is equal to mode 3, except that the information of duration of the ramps of the second triangular signal is also presented.

MTI Mode

The MTI mode, selectable in the part of *Configuration Parameters*, allows us to discard all those targets that are static. The drop-down menu is presented as follows:

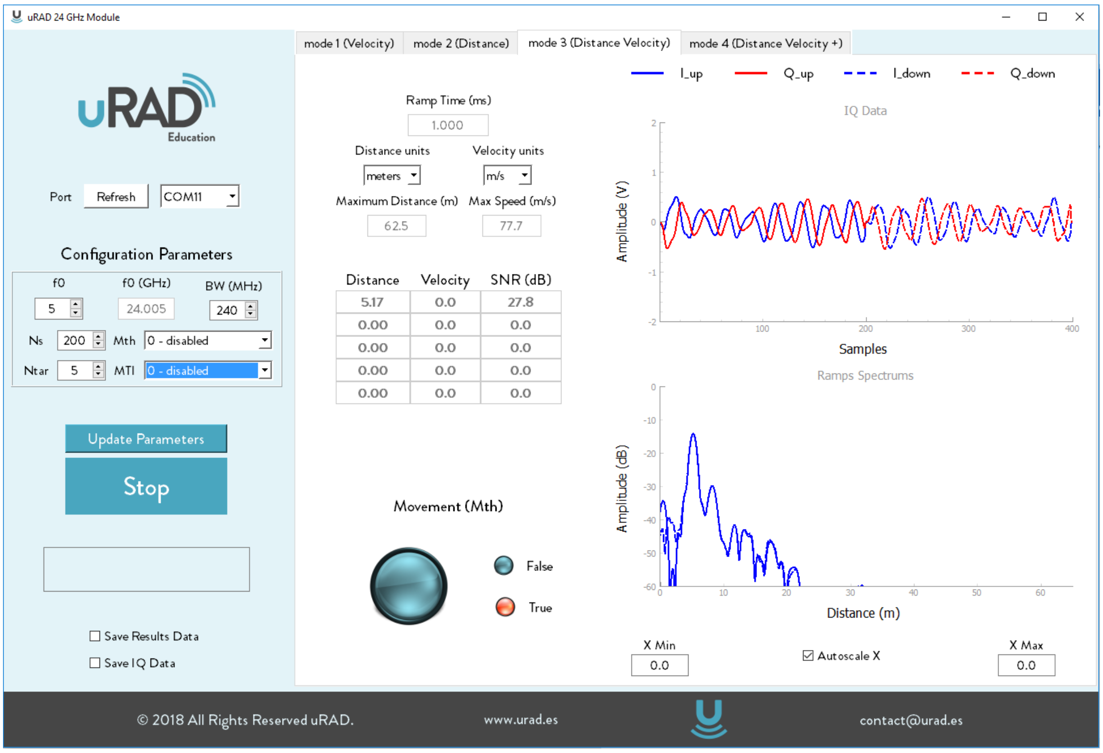
This mode only makes sense in modes 2, 3 and 4, since mode 1 is only for detecting the velocity of moving targets and then, by default, static targets are not detected.

We see below an example of using the MTI mode. In this scenario, there is a wall at 5 meters from uRAD, which remains static pointing to the wall.

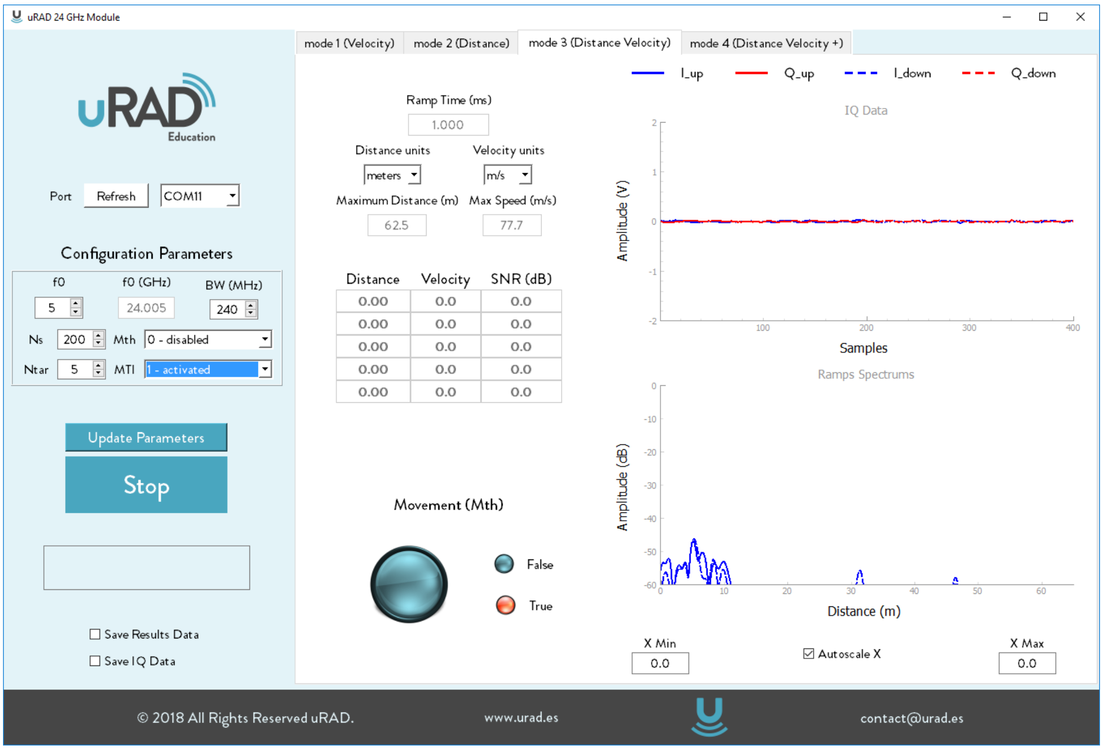
With the MTI mode disabled, the signal reflected in the *IQ Data* graph is perfectly visible and the peak in the spectrum appears. The results table also shows the distance up to the wall.

On the second capture, the MTI mode has been activated. In this case, the signals I and Q have disappeared and the spectrum does not show any peak, so the wall is no longer detected and does not appear in the results table.

MTI mode disabled



MTI mode enabled



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, not withstanding 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.