

INSTALLATION MANUAL

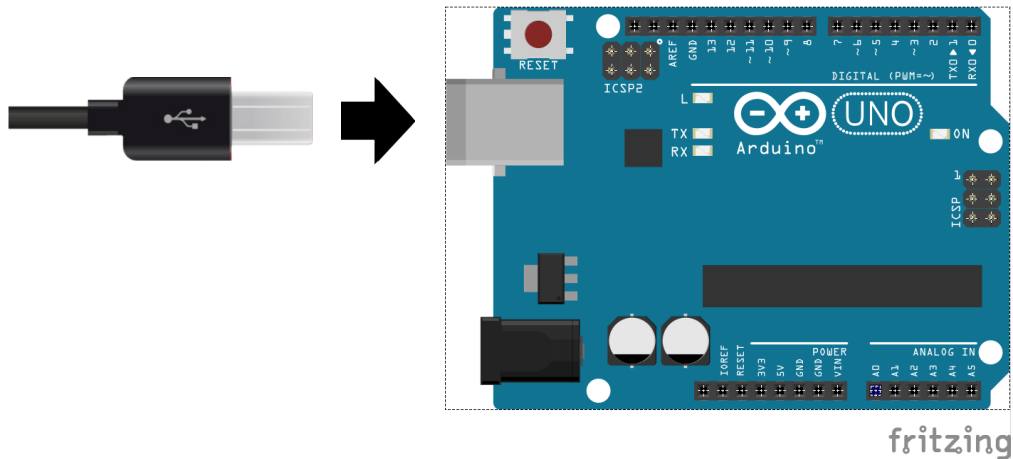
Table of Contents

Client setup.....	2
Server setup.....	2
Software compilation.....	3
Configuration.....	5
Logging.....	6
Sensors.....	7
Circuit diagram.....	8
Breadboard schematic.....	9

Client setup

To setup the device, place it at the desired location.

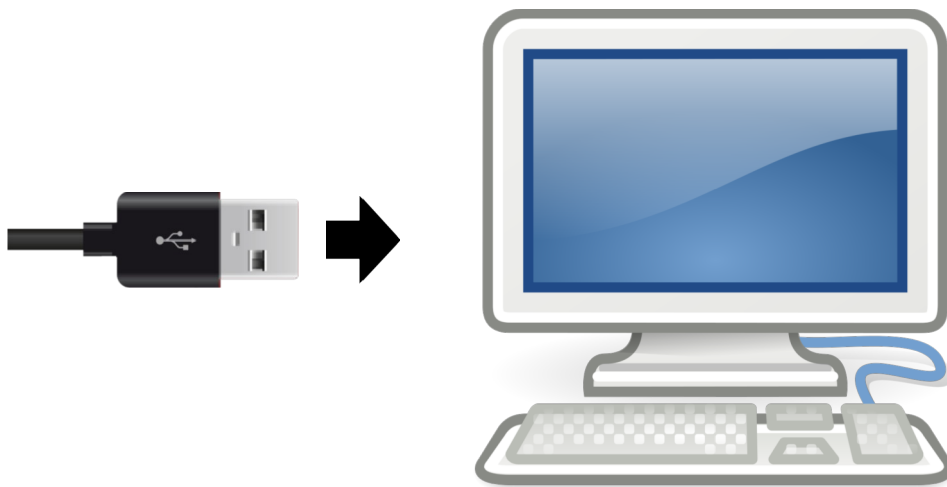
Plug in the USB cable into the device as pictured below:



Server setup

Plug in the other end of the USB cable into the device that should run the server:

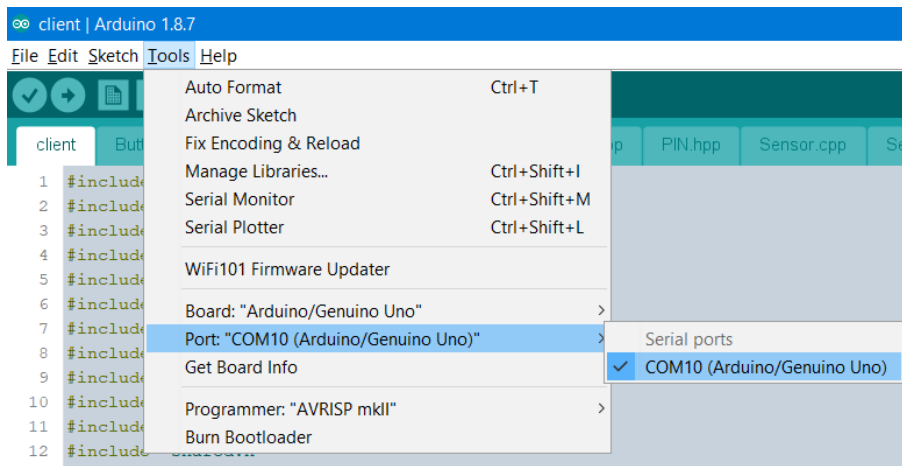
Test and make sure the server device is connected to the internet.



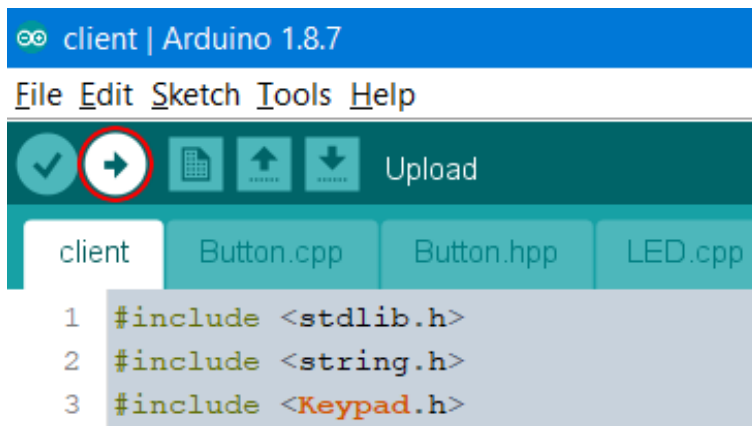
Software compilation

- To compile the software, you will need the following tools:
- Terminal/command prompt – Operating systems usually have this pre-installed
- GNU make/MinGW - <https://sourceforge.net/projects/mingw/>
- Arduino IDE - <https://www.arduino.cc/en/main/software>

Once downloaded and installed, run *Arduino IDE* and open *client.ino* inside the *client* subdirectory from the source code's root directory. Make sure “**Arduino/Genuino Uno**” is selected in **Tools > Board** and **Tools > Port** has the correct port selected in which the client device is connected:



Now press upload:



When the client has uploaded successfully, open a terminal/command prompt and navigate to the source code's root directory.

From here navigate to the *server* subdirectory.

Run *make* command:

```
$ make
gcc -pedantic -Wall -Wextra -Wconversion -g -Iinc -c ../client/crc.c -o obj/crc.o
gcc -pedantic -Wall -Wextra -Wconversion -g -Iinc -c ../client/packet.c -o obj/packet.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -I../client -c src/SensorEntry.cpp -o obj/SensorEntry.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -I../client -c src/SensorTable.cpp -o obj/SensorTable.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -I../client -D__USE_MINGW_ANSI_STDIO=1 -c src/main.cpp -o obj/main.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/misc.cpp -o obj/misc.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/UserTable.cpp -o obj/UserTable.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/Logger.cpp -o obj/Logger.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/Generic.cpp -o obj/Generic.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/UserEntry.cpp -o obj/UserEntry.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -D__USE_MINGW_ANSI_STDIO=1 -c src/SerialPort.cpp -o obj/SerialPort.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/LogEntry.cpp -o obj/LogEntry.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/LogTable.cpp -o obj/LogTable.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/Table.cpp -o obj/Table.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/Entry.cpp -o obj/Entry.o
g++ -pedantic -Wall -Wextra -Wconversion -g -Iinc -c src/RegexAssembly.cpp -o obj/RegexAssembly.o
g++ obj/crc.o obj/packet.o obj/SensorEntry.o obj/SensorTable.o obj/main.o obj/misc.o obj/UserTable.o obj/Logger.o obj/Ge
try.o obj/RegexAssembly.o -o bin/gad.exe
```

Once completed, you can now start the server by either using *make run* command:

```
$ make run
./bin/gad.exe
Sensors successfully loaded 2 entries
Users successfully loaded 5 entries
*** Connection successfully established on serial port 'COM10'
```

Or just run it directly: *./bin/gad.exe*:

```
$ ./bin/gad.exe
Sensors successfully loaded 2 entries
Users successfully loaded 5 entries
*** Connection successfully established on serial port 'COM10'
```

Configuration

The server needs to have users configured. These can be defined in the *users.dat* file in *server* root directory. The format of a user should be written in a CSV format

(https://en.wikipedia.org/wiki/Comma-separated_values)

The table below describes each field:

userID;pincode;username;tagID;status;reserved

<i>userID</i>	ID of the user (digits only)
<i>pincode</i>	PIN code of the user (4-6 digits)
<i>username</i>	Name or alias of the user (letters only, A-Z, upper or lowercase)
<i>tagID</i>	The ID of the user's tag, if the user has one, otherwise specify -1 (digits only)
<i>status</i>	Current user status: <ul style="list-style-type: none">• 1 – User is active and can log in anytime• 2 – User is currently inactive and is unable to log in• 3 – User is blocked and is unable to log in
<i>reserved</i>	Not a currently used field, this can be ignored

Examples:

1;KalleAnka;1234;101;1;

2;AlexanderLukas;03459;-1;3;

5;JoakimVonAnka;985230;105;2;

NOTE: By default, the *root* user is present and active with the PIN code *0000*.

This code should be changed as soon as possible.

Logging

There are two types of log files located in the *server* root directory.

First one is named *system.log* and has CSV (see configuration) values of users interacting with the alarm central. The file has the following layout:

index;date;userID;reserved;message

<i>index</i>	The log entry index
<i>date</i>	Time and date of the event
<i>userID</i>	ID of the interacting user. This field has a value of -1 if the user cannot be determined (For example: If the user typed the wrong PIN code)
<i>reserved</i>	Not a currently used field, this can be ignored
<i>message</i>	A message text describing the type of event

Examples:

0;1970.01.01 00:00:00;0;;User login successful

1;2018.12.24 01:15:32;3;;Alarm activated to mode A

2;2019.01.06 16:20:42;-1;;Failed login attempt

There is also a sensor status file for writing/reading current sensor statuses. This file is named *sensors.dat* and is automatically updated whenever a status changes and holds ID and Status fields described below:

ID;status

<i>ID</i>	ID of the sensor
<u><i>status</i></u>	Current status <ul style="list-style-type: none">• 0 – Sensor is closed/not triggered• 1 – Sensor is open/triggered• 2 – Sensor registered a malfunction and may need maintenance

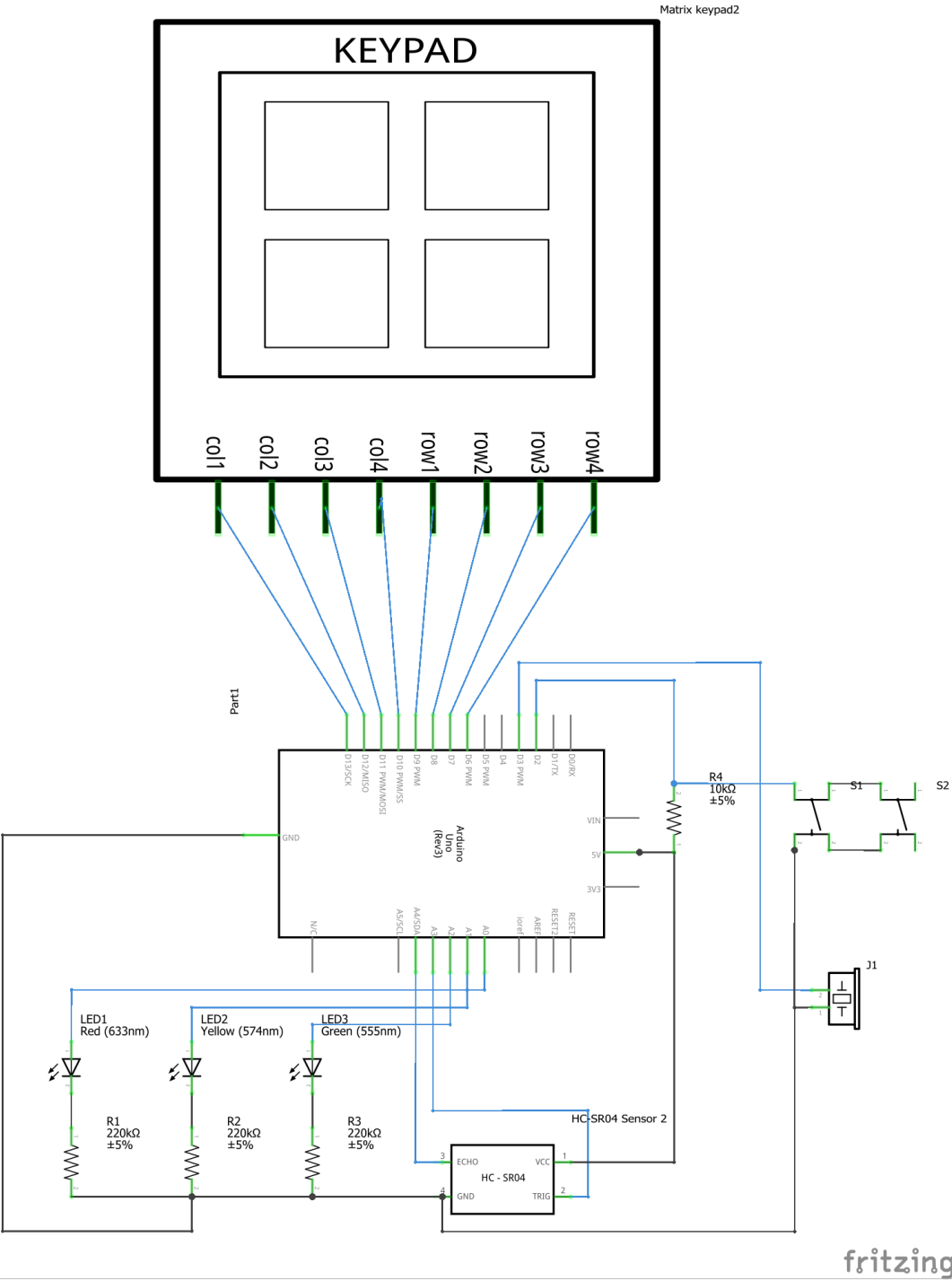
Sensors

There's two types of sensors: **Magnetic switches** and **sonic motion sensor**.

Magnetic switches should, for example be installed on doors and windows or any other openable/breakable entry point to the protected area. They should be placed close enough to each other to make contact. When the sensors are active and lose contact with one another, the alarm will trigger.

Motion sensors should preferably be placed aloft, facing a plain surface, wall or door of what would be a possible entry point for a trespasser. Having several object with different distances from each other in front of the sensor may trigger a false positive alarm. When motion sensor is activated, it scans the distance to the target surface and keeps that value for future comparisons. Whenever the distance is changed i.e. something has moved in-between the sensor and the surface, the alarm will trigger. Therefore, be cautious to have moving entities in the area as this sensor is activated. The motion sensor has a distance threshold of 0.5 meters (1.64 ft). If the motion sensor would break from a defect or a distance read failure due to getting the initial monitoring distance at activation process, the alarm will also trigger. This will happen if the duration of faulty readings reaches 1 second and above (5 seconds at the startup process).

Circuit diagram



Breadboard schematic

