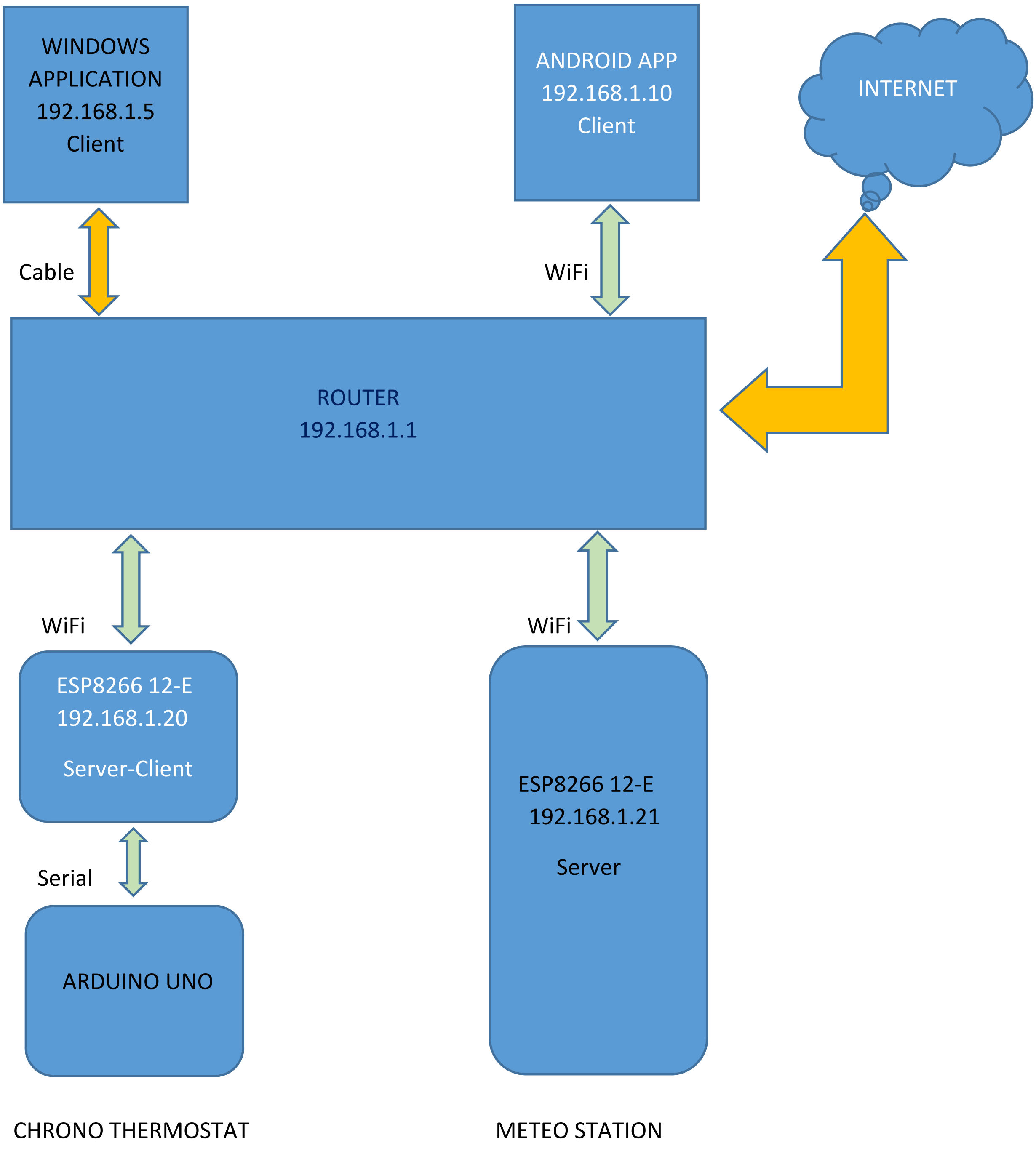
**Project:** **WEEKLY CHRONOTHERMOSTAT**

**Reference technologies:** **IOT - ESP8266 + Arduino**

The project includes the following devices:

- N° 1 Arduino Uno with ESP8266 on board (Weekly ChronoThermostat)
- N° 1 ESP8266-12E (External Meteo Station)
- N° 1 Custom Arduino Shield
- Several sensors for environmental data
- N°1 Android app
- N°1 Windows application
- All the projects al developed with Visual Studio + Visual Micro plug-in

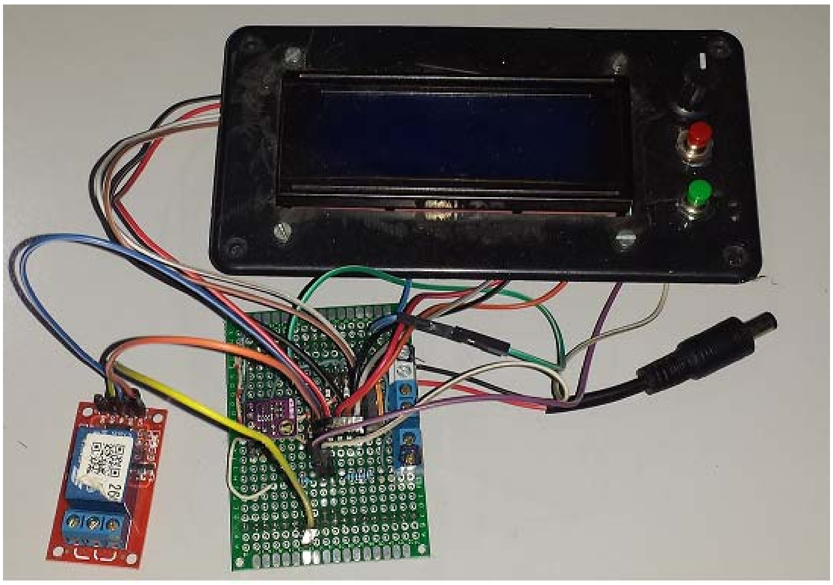
All of the systems are connected to the local area through the router. The ESP8266's are configured as Station.



## 1.1 WEEKLY CHRONO THERMOSTAT

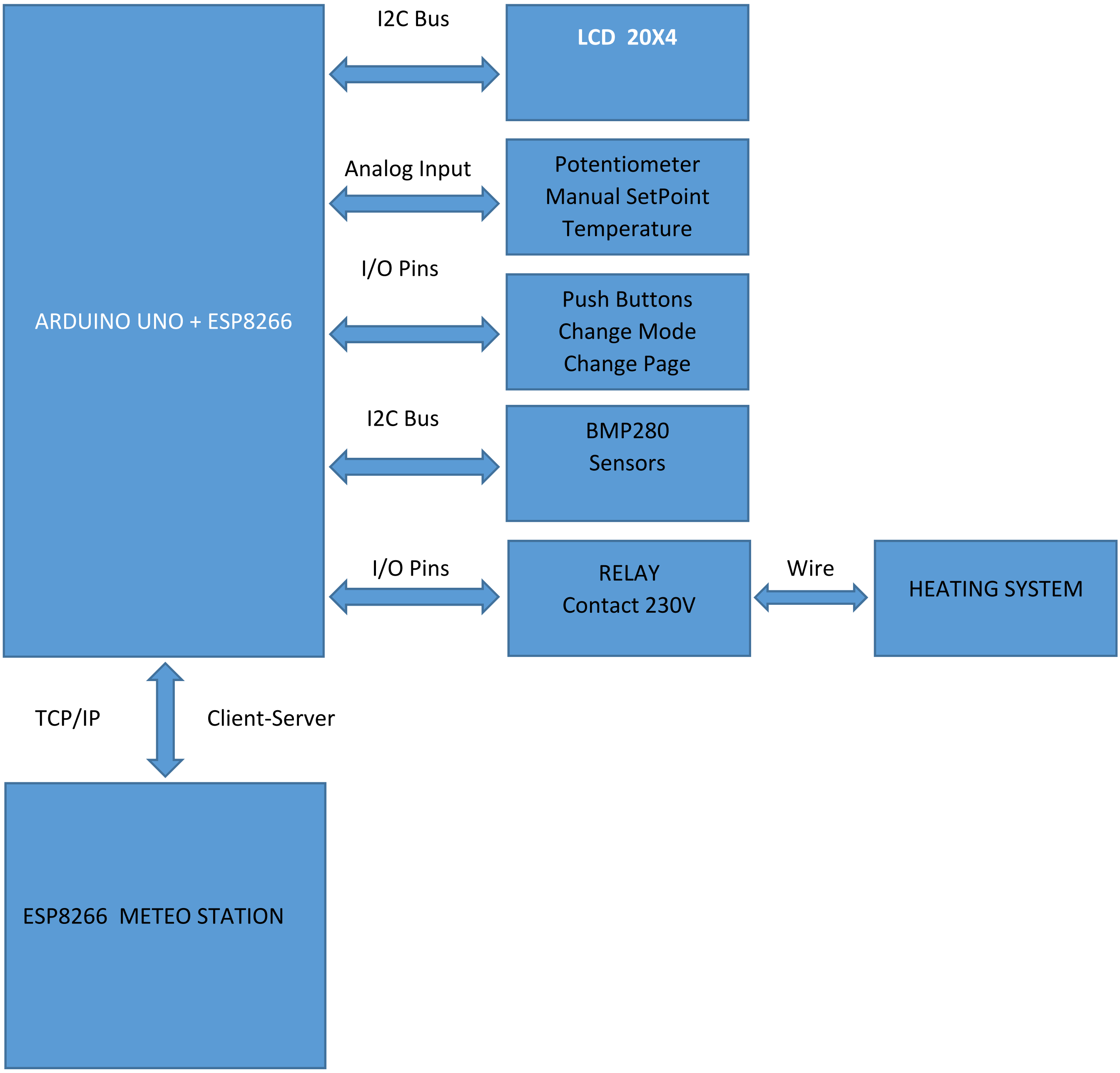
The weekly thermostat is a common device, which controls the eating system in your house. There are many of commercial controls like it. This project wants to include one of them, one external meteo station, external control by using a windows application and android app. The weekly chrono thermostat is composed by the following devices and technologies. The most important innovation is the new Arduino board which includes the ESP8266 12E V3 Lolin. The two devices are independent and we can program them by using Arduino IDE. A set of dipswitches allows us to program ESP8266 or Arduino or debug or put them in communication over the serial line. Here the picture:

The top side of the module is reserved for the ESP8266. On the bottom there is the classic Arduino Uno Rev. 3. The usb port allows to program and debug the modules, but separated. To connect all of the parts, I built a custom shield. The shield has connectors to supply all the components such as the LCD 20x4, potentiometer, push buttons, relay and more. The environmental data such as temperature, humidity and pressure are taken from the BMP280 module. The BPM280 and the LCD, are connected in I2C bus. The shield is installed on Arduino Uno-



The main power supply is 12V. An LM7805 stabilizes 5V to supply LCD, Relay and else. The Arduino Uno and his ESP8266 are directly supplied by 12V.

Let’s see the schematic of the Weekly Chrono Thermostat:



The Arduino Uno is connected to the LCD to show the parameters such as the temperature, the pressure and more. The push button Change Mode changes the working mode between local and remote. In local mode is the potentiometer to set the set point of temperature desired. In remote mode is the client that send to the Arduino the set point and many other jobs to do. The relay supports 230V in its contact and it is connected to the eating system by using a normally closed contact.

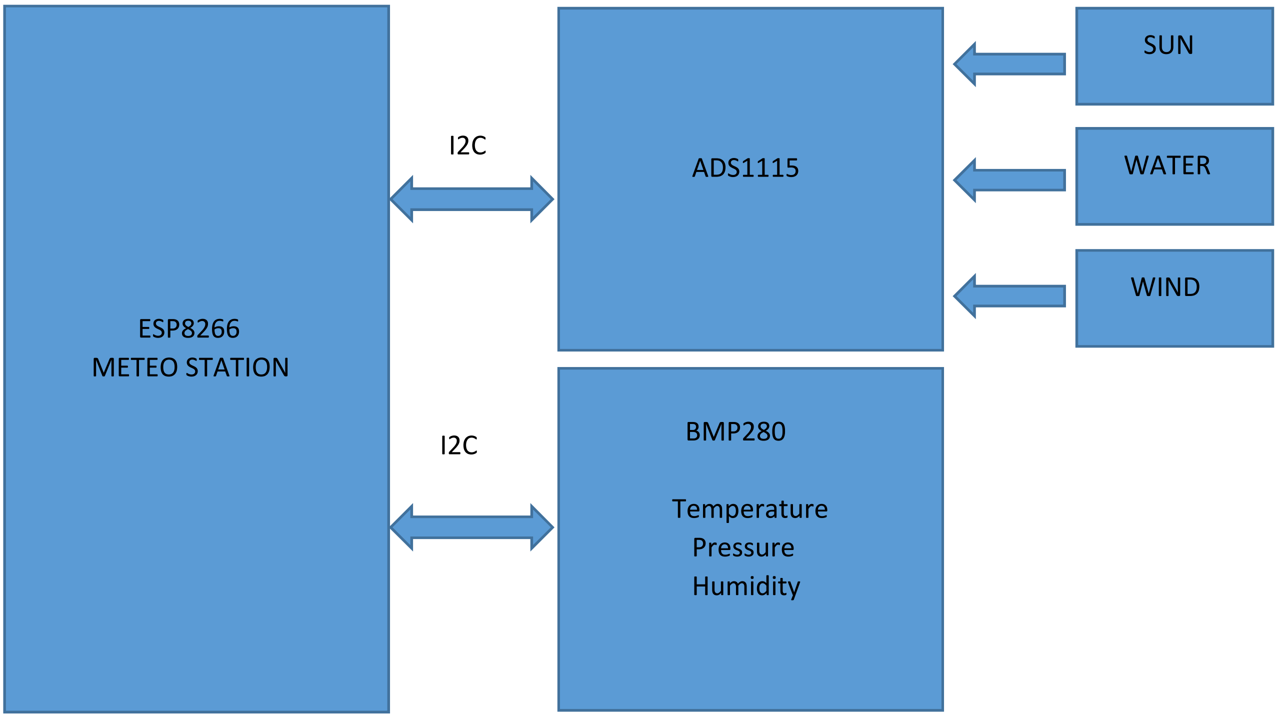
Inside the chrono thermostat, there are all the data to run a complete weekly eating control. The android client is able to manage and to send the daily settings for seven days. Each day is represented by 24 hours and three thresholds, T1, T2 and T3. The data can be saved into the EEPROM of Arduino and to reload them when the application restarts.

## 1.2 METEO STATION

The meteo station is installed externally. The board is connected to all of the sensors to acquire these values:

- External Temperature
- External Pressure
- External Humidity
- External Intensity of Sun
- External Intensity of Water
- External speed of Wind

The main controller is the ESP8266 12-E embedded microcontroller. The choice maybe to use another Arduino Uno + ESP8266. Now the only ESP8266 is enough. Because the ESp8266 has only one analog input, the project includes a chip, the ADS1115 with 16 Bits analog input in I2C bus. Moreover, there is a BMP280 chip already presented above.



The ESP8266 embedded into Arduino Uno is the core of the communication. It is connected to the Arduino Uno chrono thermostat in serial way and it is connected to the meteo station as Tcp/Ip client. The meteo station is a TCP/IP server. The core of the communication gets the external data every 2 seconds alternately with the internal data. Every 4 seconds all the environmental data are refreshed.

## 1.3 WINDOWS APPLICATION

To communicate with the core of communication, the ESP8266 embedded into Arduino Uno, it is possible to use each TCP/IP client. The protocol for the communication is the W-PROTOCOL, a Modbus-like binary protocol that supports all the commands for this project.

## 1.4 ANDROID APP