NAME:
ECHEREBOR EFE CHRISTIAN.

MATRIC NO:
18/ENG05/014.

DEPARTMENT:
MECHATRONICS ENGINEERING.

COURSE CODE:
ENG 224.
COURSE:
STRUCTURED

COMPUTER PROGRAMMING

A. If you want your farming to be effective you will need your soil quality to be at it's best, like the temperature of the soil must be accurate for whatever you want to plant, for my software to be able to check soil temperature it will need sensors, Arduino is a perfect example to create a software, The Arduino Nano was used as a microcontroller for the Node of sensor network, with the following dimensions: 18 × 45 mm, Flash memory of 32 Kbytes, 8 analogue inputs/outputs, 22 digital inputs/outputs and consumption 19 mA in 5 V. A two-way communication between Nodes and Gateway was performed by radio NRF24L01. The digital humidity and temperature sensors were SHT20 and DHT22 (this last one in two versions). In addition, the FC-28 analogue humidity resistive sensor was used, and the Real Time Clock (RTC) DS1302 was also used to establish temporal synchronism among readings. The system was powered by a 9-V rechargeable battery, which is charged by an LM317 composite circuit, diodes, resistors and three solar panels of 5 V and 200 mA each. Figure 1 shows the Sensor Node circuit. The Gateway is connected to Sensor Nodes on NRF24L01 radio at a maximum distance of 100 meters. The Sensor Nodes and Gateway store the smallest path for data transmission after initialization in its non-volatile memory. When a Sensor Node is connected, a start up package is sent by radio. If the package is received directly by the Gateway, it will provide an identification number for the requestor and will return a response to the Node that began the dialog via radio. If the Sensor Node is physically far from the Gateway (more than 100 m) and cannot communicate directly, one or more Sensor Nodes may act as intermediaries to send the data packets. The start up package identifies the sensor and the type of data it

provides, humidity, temperature, or voltage. This allows that a Sensor Node can be inserted into network at any time.

A radio network can consist of up to 254 different radio nodes and each radio node can transmit data for up to 254 sensors. In this case, it is possible to manage data of up to 64,516 sensors in a single radio network. There is also the possibility of creating parallel radio networks with 126 channels available for NRF24L01. Thus, the radio meets the needs for the requested monitoring activity. The programming of Arduino boards was carried out in an available development environment offered by the board's manufacturer, in version 1.6.7 for the Windows 10 operating system. The routines for humidity and temperature sensors readings were developed using outsourced party free libraries that follow the software environment package, plus My Sensor, DHT, SHT2x and Wire libraries.

Radios were chosen in the market due to their lowest price and highest performance, which allowed communication among Arduino modules and the set of free libraries to guarantee communication among Sensor Nodes and Gateway of the mesh network. There were installation and configuration of Arduino Uno with Shield W5100 Ethernet card and digital radio cards (gateway), routines for radio communication on Arduino Nano (Sensor Node), routines to read the Real Time Clock circuits (RTC) DS1302 as well as the server to receive data from the sensor nodes through the gateway. The experiment was carried out indoors with controlled temperature, moisture and luminosity. The twenty sample units were randomly arranged, and each sample received the same amount of soil.

One of the tubes that already contained soil was taken, then, sensors connected to the Sensor Node and the 20-cm probe of TDR were inserted into this tube, which was closed at the end. Five readings were recorded on each sensor at every two hours and their average was calculated. This action was repeated for each tube. Then, in total, twenty measurements of moisture, temperature, voltage, dielectric constant and volumetric humidity were collected.

Among the readings of measurements from one tube to the other, sensors were removed, left under shelter and at room temperature for two hours so that the readings in other tubes with the same sensors did not undergo the effect of previous measurements and were resilient, before measuring them again. The hardware designs

of Gateway and Sensor Node have worked properly for the requirement to use free technologies and the platform used by Arduino confers this feature. They take up little physical space, they also have low power consumption, high performance, and the price of their components are far below those presented by many other technologies. The battery system allows operation without recharging up to 36 hours and, with the sun, the photovoltaic cells can guarantee uninterrupted operation.

The applied sensors and radio had very low cost and it has been demonstrated that it is feasible to use all the devices, mainly, the resistive one. It should be highlighted that this or the other capacitive ones can replace TDR use to register humidity.

B. Hardware features

Power (USB / Barrel Jack)

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack.

Pins (5V, 3.3V, GND, Analog, Digital, PWM, AREF)

The pins on your Arduino are the places where you connect wires to construct a circuit (probably in conjunction with a breadboard and some wire. They usually have black plastic 'headers' that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins, each of which is labelled on the board and used for different functions.

- ***GND: Short for 'Ground'. There are several GND pins on the Arduino, any of which can be used to ground your circuit.***
- ***5V & 3.3V : As you might guess, the 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.***
- ***Analog : The area of pins under the 'Analog In' label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analogue sensor (like a temperature sensor) and convert it into a digital value that we can read.***
- ***Digital: Across from the analogue pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).***

- ***PWM: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM).***
- ***AREF: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analogue input pins.***

Reset Button

The Arduino has a reset button, Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn't repeat, but you want to test it multiple times.

Power LED Indicator

Just beneath and to the right of the word "UNO" on your circuit board, there's a tiny LED next to the word 'ON', this LED should light up whenever you plug your Arduino into a power source. If this light doesn't turn on, there's a good chance something is wrong. Time to re-check your circuit!

TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. In our case, there are two places on the Arduino UNO where TX and RX appear -- once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs (12). These LEDs will give us some nice visual indications whenever our Arduino is receiving or transmitting data (like when we're loading a new program onto the board).

Main IC

The black thing with all the metal legs is an IC, or Integrated Circuit, think of it as the brains of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the AT mega line of IC's from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC's, reading the datasheets is often a good idea.

Voltage Regulator

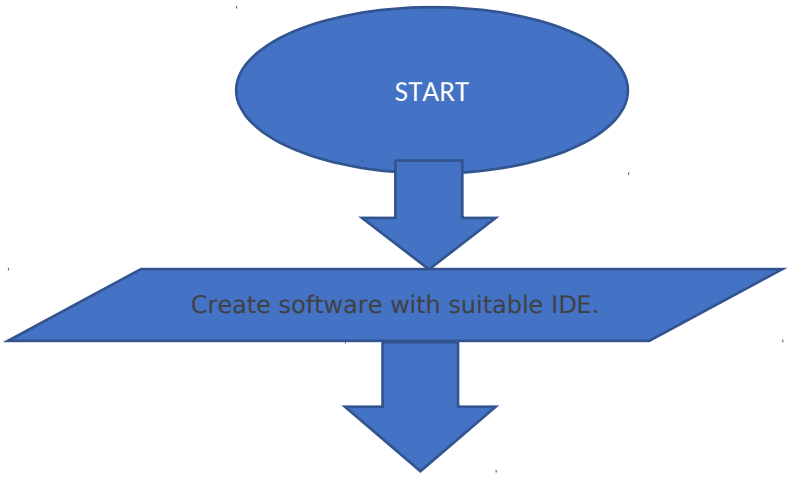
The voltage regulator is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it's for. The voltage regulator does exactly what it says -- it controls the amount of voltage that is let into the Arduino board. Think of it as a kind of gatekeeper; it will turn away an extra voltage that might harm the circuit. Of course, it has its limits, so don't hook up your Arduino to anything greater than 20 volts.

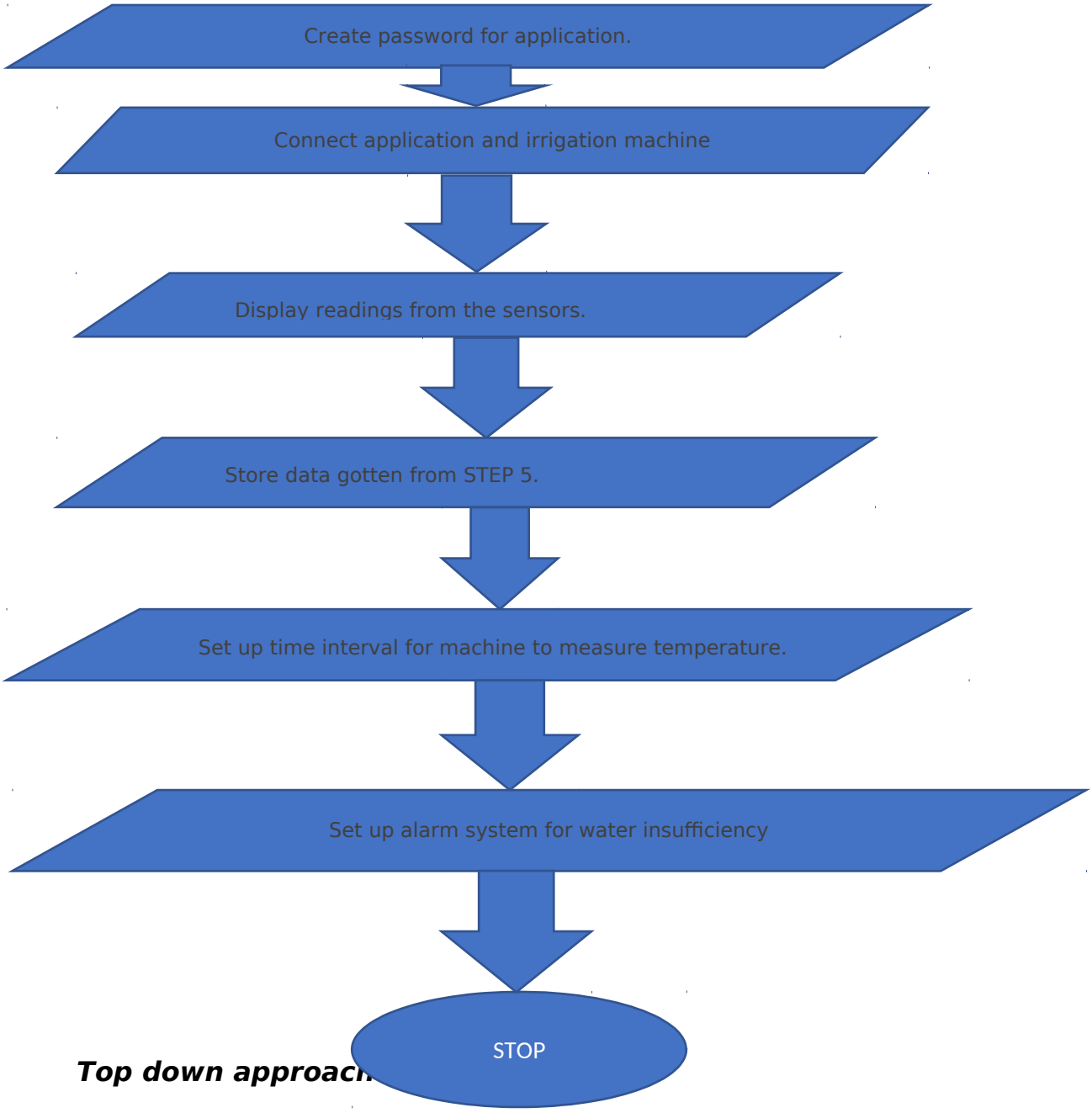
SOFTWARE FEATURES

- ***Microsoft Windows XP with SP2, Windows 7, Windows 8/8.1 and Windows 10 operating system.***
- ***Microsoft .NET Framework 3.5 or higher.***
- ***Intel Pentium / AMD Athlon processor or equivalent running at 1 GHz or more.***
- ***512 MB RAM (1 GB RAM recommended).***
- ***10MB free hard drive space or more (only for PROGRAMINO IDE for Arduino).***

C. ***Step 1: Start.***
Step 2: Create software with suitable IDE.
Step 3: Create password for application.
Step 4: Connect application and irrigation machine.
Step 5: Display readings from the sensors.
Step 6: Store data gotten from STEP 5.
Step 7: Set up time interval for machine to measure temperature.
Step 8: Set up alarm system for water insufficiency.
Step 9: Stop.

FLOWCHART





Top down approach