We will use **CodeVisionAVR** software for writing our code and **SinaProg** software for uploading our code to Atmega8 using USBASP programmer.

You can download these softwares from the given links:

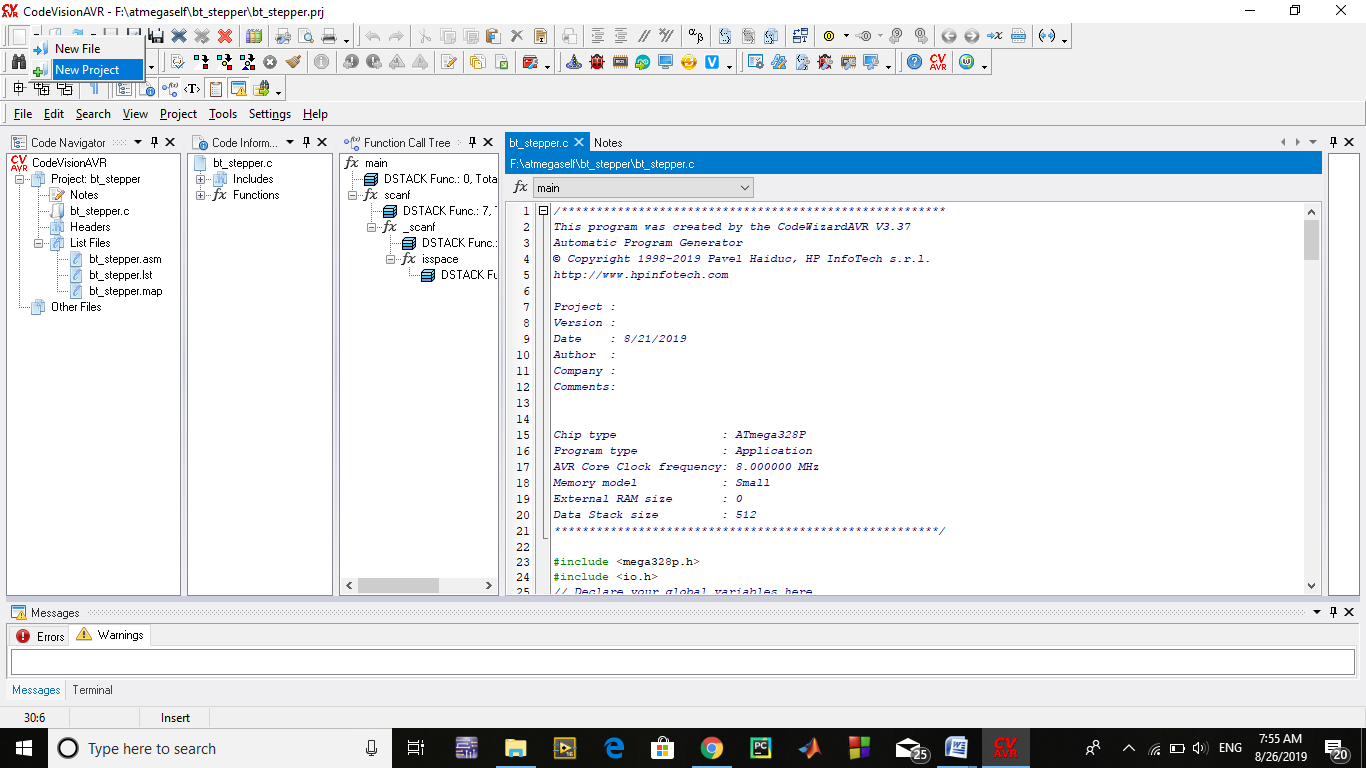
**CodeVisionAVR**: <http://www.hpinfotech.ro/cvavr_download.html>

**​**

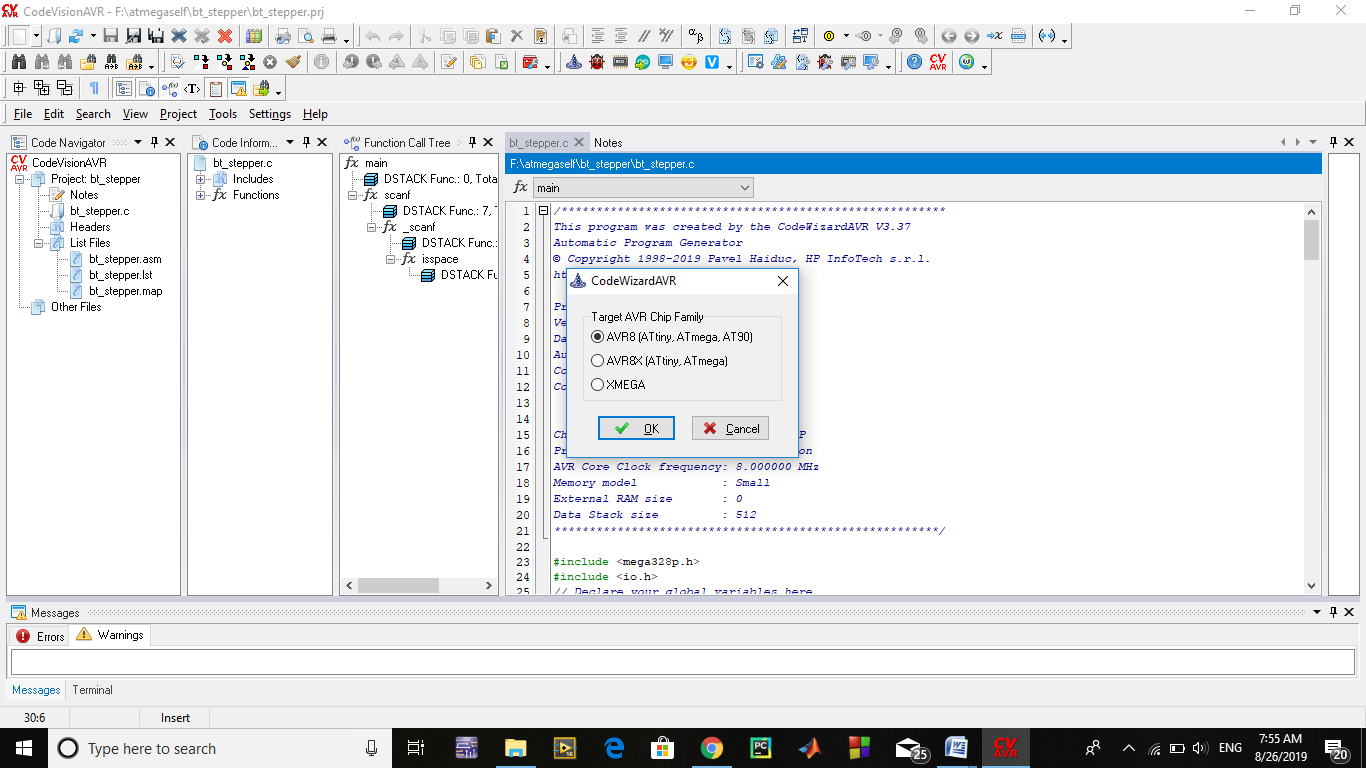
Creating the Project for Atmega 8 using CodeVision:

After installing these softwares follow the below steps to create project and writing code:

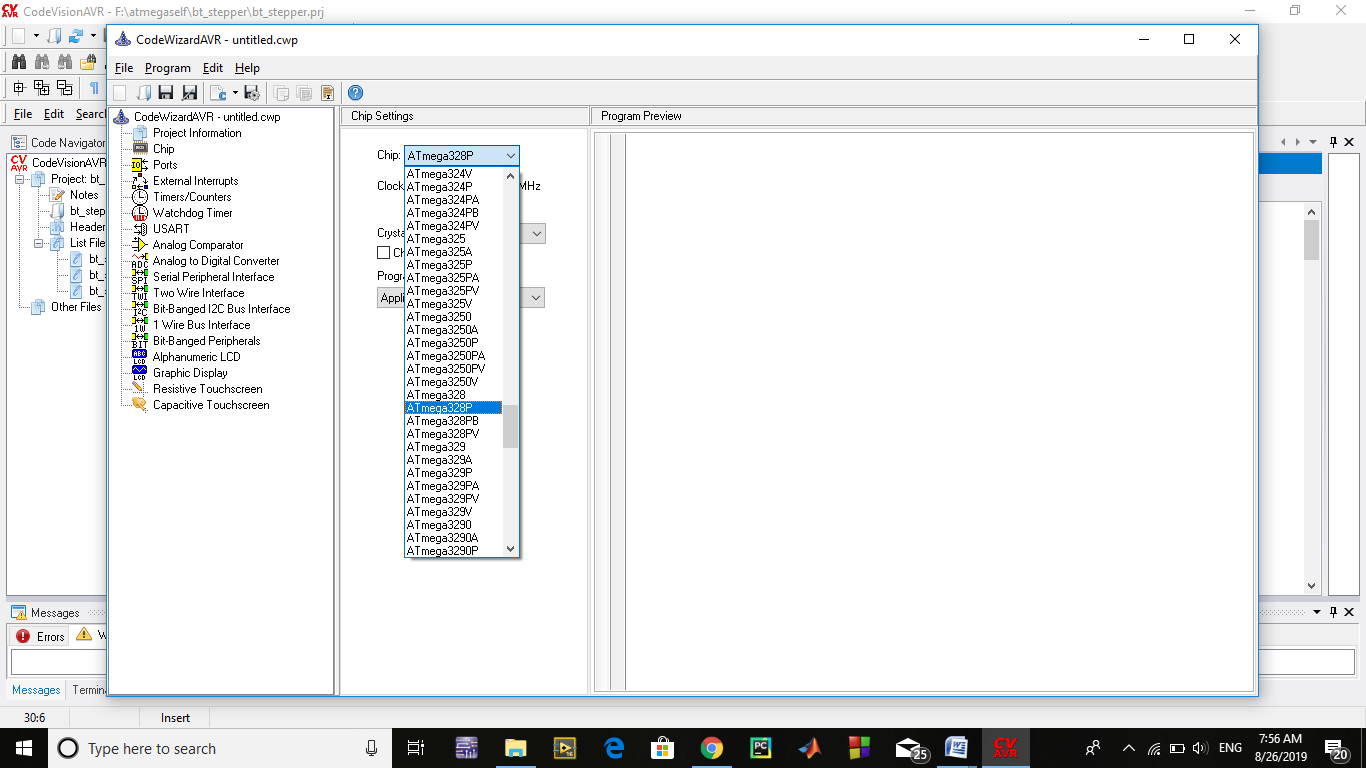
**Step 1**. Open CodeVision Click on *File -> New -> Project*. Confirmation Dialogue box will appear. Click On Yes



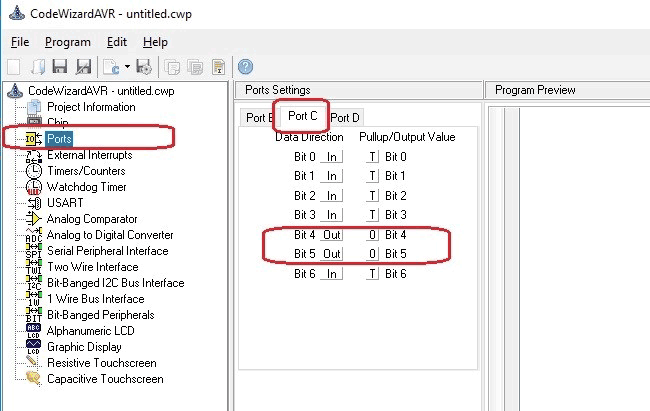
**Step 2.**  CodeWizard will open. Click on first option i.e.*AT90*, and click OK.



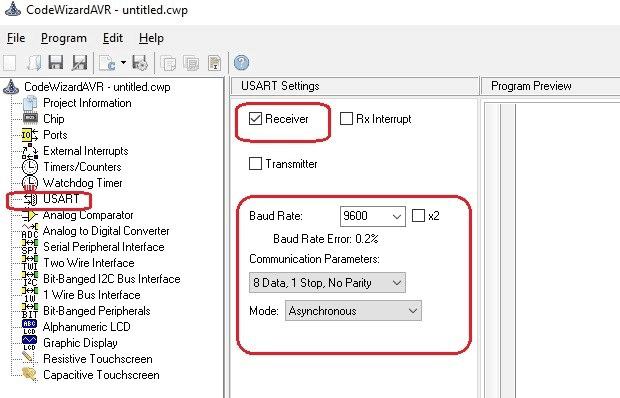
**Step 3.** Choose your microcontroller chip, here we will take Atmega8 as shown.



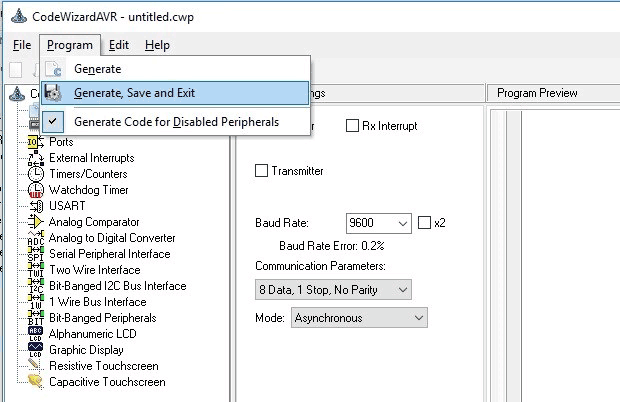
**Step 4.** Click on *Ports*. In our project, we will be using Port C4 and C5 for led interfacing. So, make Bit 4 and bit 5 as output by clicking on it. As shown below:



**Step 5.** We will use USART for Rx and Tx. So, click on USART option and click on Receiver option and leave rest as it is.

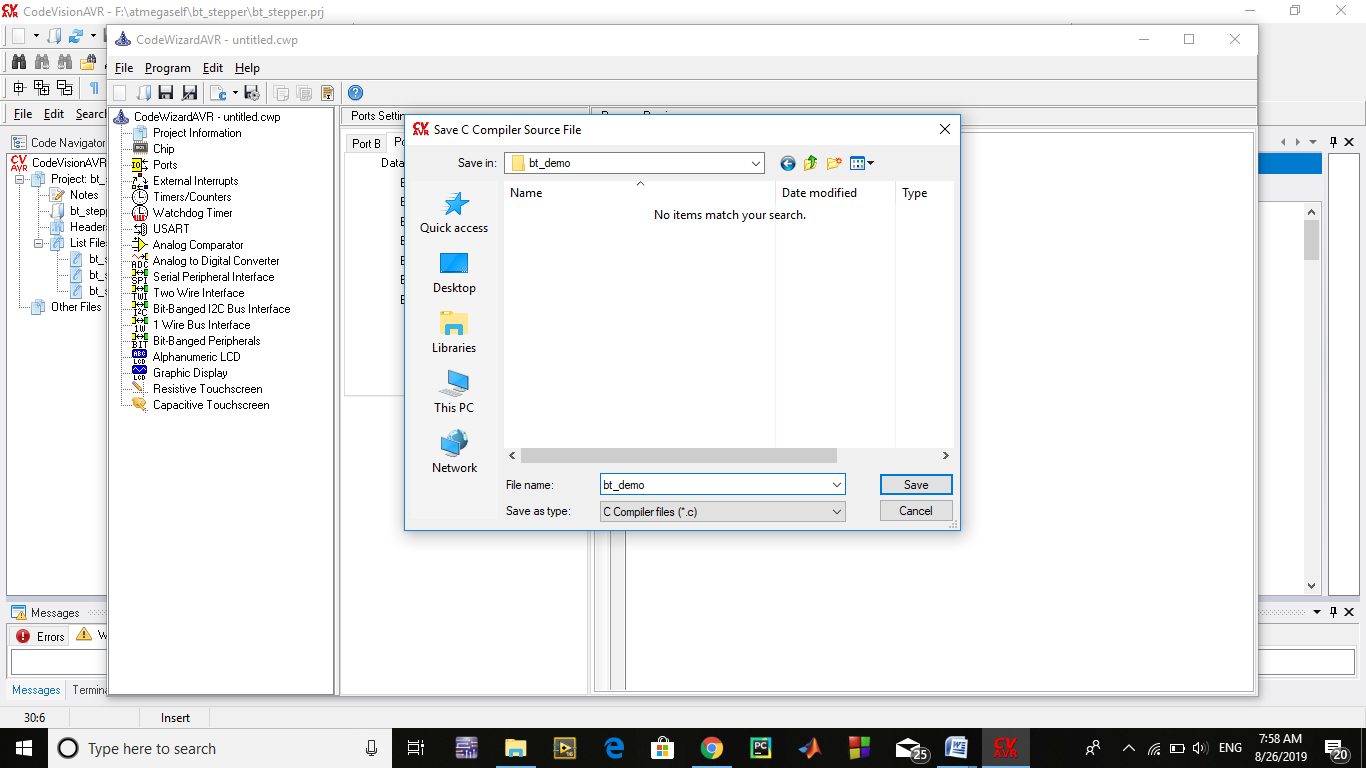


**Step 6.** Click on *Program -> Generate*, *Save and Exit*. Now, more than half of our work is completed



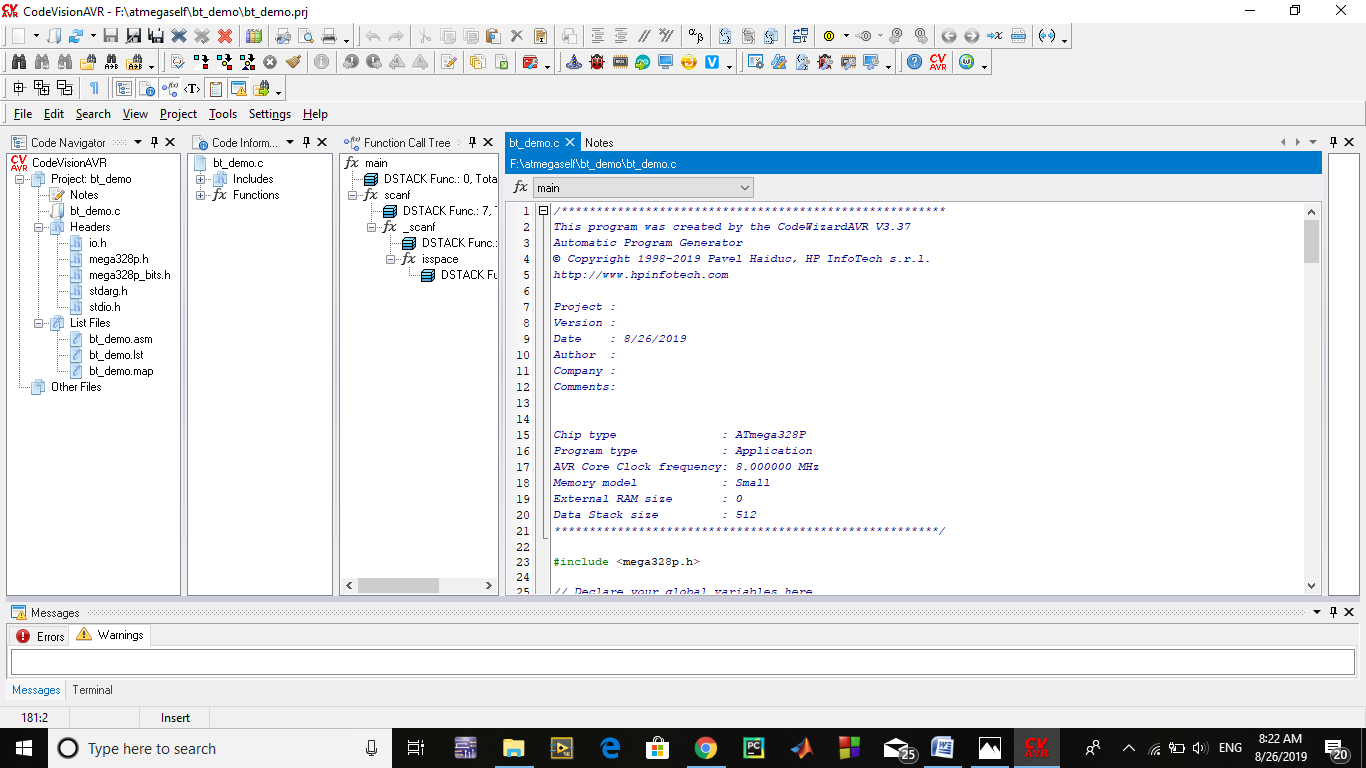
**Step 7**. Make a New folder on desktop so, that our files remains in folder otherwise it we will be scattered on whole desktop window. Name your folder as you want and I suggest use the same name to save program files.

We will be having three dialogues box one after other to save files.



Do the same with other two dialogue boxes which will appear after you save the first.

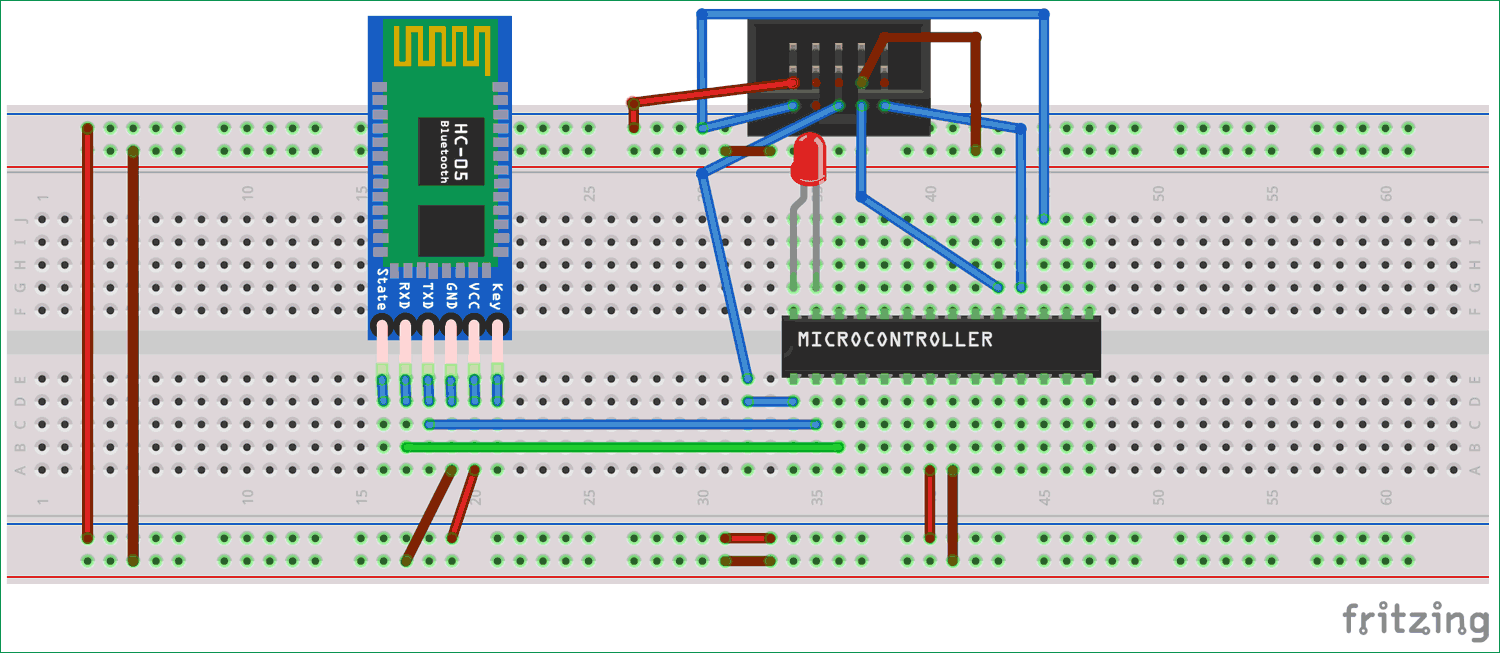
Now, your workspace looking like this.



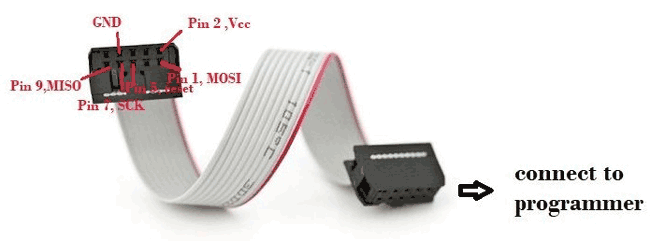
Our most of the work is completed with the help of the Wizard. Now, we have to write only few lines of code to interface the bluetooth module and control the LED.

Circuit Diagram:

Circuit diagram for **interfacing Bluetooth HC-05 with AVR** is given below.

[](https://circuitdigest.com/fullimage?i=circuitdiagram_mic/Circuit-diagram-for-interfacing-Bluetooth-HC-05-with-Atmega8_0.png)

Hookup the one side of FRC cable to USBASP programmer and other side will connect to SPI pins of microcontroller as described below:



1. Pin1 of FRC female connector -> Pin 17 ,MOSI of Atmega8
2. Pin 2  connected to Vcc of atmega8 i.e. Pin 7
3. Pin 5 connected to Reset of atmega8 i.e. Pin 1
4. Pin 7 connected to SCK of atmega8 i.e. Pin 19
5. Pin 9 connected to MISO of atmega8 i.e. Pin 18
6. Pin 8 connected to GND of atmega8 i.e. Pin 8

Connect the remaining components on the breadboard as per circuit diagram**.**

Code and Explanation:

**Complete AVR code with Demonstration Video** is given at the end of the article.

Here we have declared a variable in ***void main*** function for storing incoming character from Bluetooth module.

**#include <io.h>**

**// Declare your global variables here**

**// Standard Input/Output functions**

**#include <stdio.h>**

**void main(void)**

**{**

**char var; // Declare your local variables here**

Rest of the code is easy and can be easily understandable. Now, come on the last lines of code where you will find a ***while* loop.**Our main code will be in this loop. Here we are continuously checking the incoming character from the Bluetooth module and turning the LED On or off accordingly.

**while (1)**

**{**

**scanf("%c",&var); //this function is to used to check any character coming from our android app .**

**if (var == 'a') // We will send ‘a’ from Bluetooth Terminal to ON the LED and ‘b’ to OFF the LED**

**{**

**PORTC.5 = 1;**

**PORTC.4 = 0;**

**}**

**if (var == 'b')**

**{**

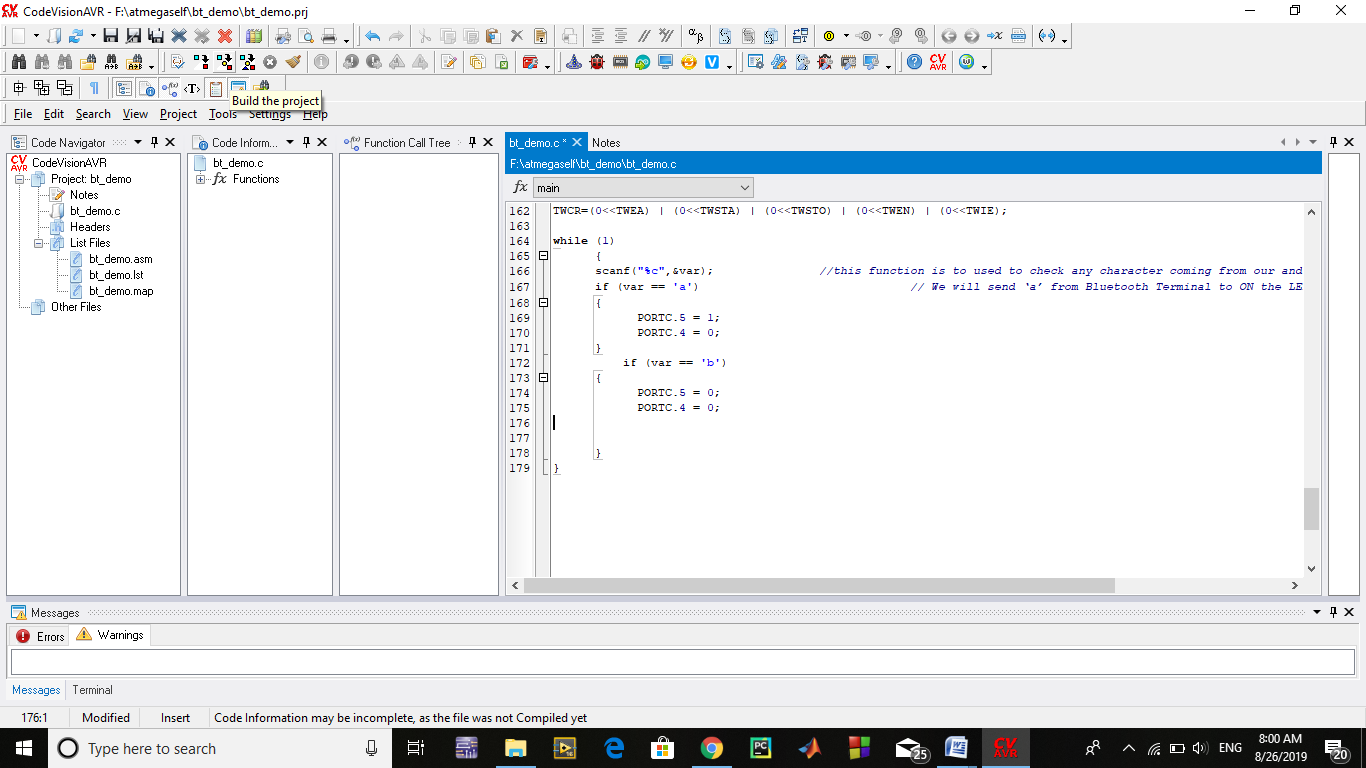
**PORTC.5 = 0;**

**PORTC.4 = 0;**

**}**

**}**

Our code is completed. Now, we have to build our project. Click on *Build the project* icon as shown.

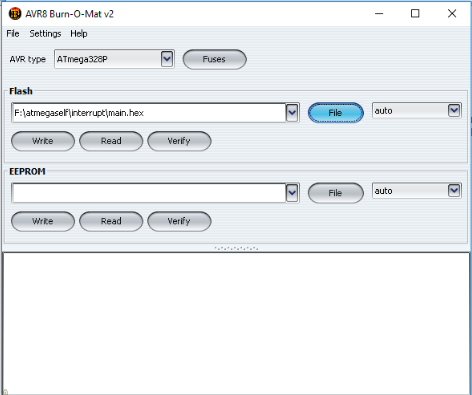


After building the project, a HEX file is generated in the *Debug-> Exe* folder which can be found in the folder you make previously to save your project. We will use this HEX file to upload in atmega8 using *AVR-BURN-O-MAT* software.

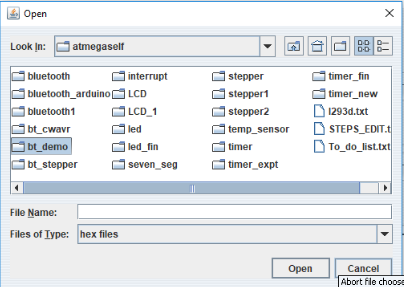
FOR UPLOADING HEX-FILE

Open Burn-o-Mat

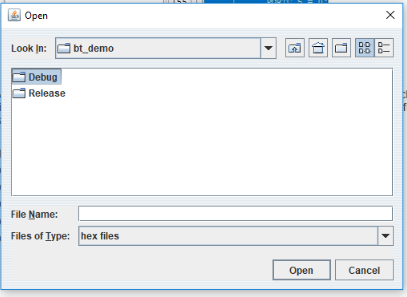
Click on File



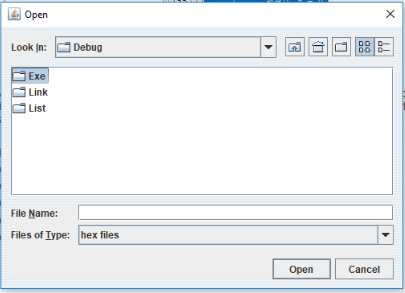
Open your folder



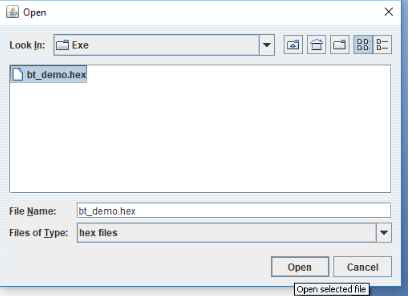
Open debug



Open Exe



Select .hex file and open



Click on write

