What are Hex and .ino files

And the difference

.ino Files

- Source code written in arduino
    - A sketch that contains commands to be compiled/assembled into a program - in our case, a HEX file.
- High level programming language
    - Code that is closer to human language
- Arduino code written in C/C++
- As it is source code, we can edit and update the code using the Arduino IDE.

.ino Files

Why use .ino Files?

- Allows us to edit and update the code using a high level language in the Arduino IDE.
- Allows us to easily upload and test the code.

Disadvantages:

- All libraries and referenced code is required for the program to run.
- The file is generally larger
- Slower upload speed as the code must be verified and compiled everytime.

.hex Files

- Object code created by Arduino
  - When we compile the .ino file, we get the object code (or hex file(
- Low level programming language
  - Machine code such as binary or Hex
- Written in Hexadecimal to convey binary data to a microcontroller
- As it is compiled object code, we cannot edit it using the Arduino IDE

.hex Files

Why use .hex Files?

- Fast and efficient upload
- File is smaller in size.
- Allows programs to be distributed without the worry of missing libraries - as it is compiled object code

Disadvantages:

- The code cannot be adjusted
- Harder to understood as it is written in hexadecimal/object code

So what does that mean

Some major key terms

Compiler

- A compiler scans through entire software
- Creates object code (the HEX file in our case) that the computer can execute directly.
- Once the object code is created, the Source code is not required for the program to run.
- Faster as once the object code is compiled, it can be executed directly

Interpreter

- An Interpreter converts the instructions line by line.
- Directly executes the source code instructions without previously converting them to object code.
- Source code and interpreter are required.
- Generally runs slower as everytime the program is ran, it needs to be interpreted.

High level languages Vs Low Level Languages

- Easier to understand
  - As it is closer to human language
- Requires a compiler or interpreter to translate it into low level language
- Can run on a variety of platforms
- Larger file sizes but less code required to do more operations (less lines do more code)

- Harder to understand
- Smaller file sizes but longer written code (more lines)
- Requires a an assembler to

To recap

Now that we know the key terms!

.ino Files

- Source code written in arduino
  - A sketch that contains commands to be compiled/assembled into a program - in our case, a HEX file.
- High level programming language
  - Code that is closer to human language
- Arduino code written in C/C++
- As it is source code, we can edit and update the code using the Arduino IDE.

.hex Files

- Object code created by Arduino
  - When we
- Low level programming language
  - Machine code such as binary or Hex
- Written in Hexadecimal to convey binary data to a microcontroller
- As it is compiled object code, we cannot edit it using the Arduino IDE

How do we get .hex files?

Let's find out!