**INTERNET OF THINGS ASSIGNMENT RECORD**

**Subject code : BTCS-AMDS-009T**

|  |  |
| --- | --- |
| Name: | Sreerama Monisha |
| Registration Number: | FET-BAML-2022-2026-004 |
| Course: | B.Tech CSE AIML |
| Semester: | 5th |
| Faculty: | Mr. Biswajeeban Mishra &  Mr. Pritam Nanda |

|  |  |
| --- | --- |
| Remarks |  |
| Signature |  |



SRI SRI UNIVERSITY

Bidyadharpur, Cuttack, Odisha.

Index

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Date** | **Experiment/Case Study** | **Page No.** | **Remark** |
| **1** | 9-8-2024 | Prototyping and Arduino UNO R3. |  |  |
| **2** | 9-8-2024 | Encoding formats. |  |  |
| **3** | 9-8-2024 | Basic Structure of an Arduino Program. |  |  |
| **4** |  |  |  |  |
| **5** |  |  |  |  |
| **6** |  |  |  |  |
| **7** |  |  |  |  |
| **8** |  |  |  |  |
| **9** |  |  |  |  |
| **10** |  |  |  |  |
| **11** |  |  |  |  |
| **12** |  |  |  |  |
| **13** |  |  |  |  |
| **14** |  |  |  |  |
| **15** |  |  |  |  |
| **16** |  |  |  |  |

***Assignment No.:1 Date: 9-8-2024***

**Q1. *What is a Prototype?***

***What are Open source and closed source prototype platforms?***

**A1.**

**A prototype** is a basic version of a product made to test ideas and see how it works before creating the final version.

**Open Source Prototype Platforms**

* Platforms where the source code is publicly available for anyone to use, modify, and distribute.
* **Examples**:
  + **Arduino**: Easy-to-use electronics platform for interactive projects.
  + **Raspberry Pi**: Affordable, small computer for various programming and electronic projects.
  + **OpenFrameworks**: C++ toolkit for creative coding and experimentation.

**Closed Source Prototype Platforms**

* Proprietary platforms where the source code is not shared with the public, and modifications are restricted.
* **Examples**:
  + **LabVIEW**: Visual programming environment for test, measurement, and control applications.
  + **MATLAB**: Numeric computing environment for algorithm development, data analysis, and visualization.
  + **Autodesk Fusion 360**: 3D CAD, CAM, and CAE tool for product design and manufacturing.

1. **Q2. What is Arduino?**

Arduino is an easy-to-use electronics platform that helps you create interactive projects.

It consists of both a physical programmable circuit board (often called a microcontroller) and software (an IDE) that runs on your computer, allowing you to write and upload code to the board.

**Q3. *Write down the Arduino Uno R3 Key Specifications.***

**A3.**

* **Main Processor :**

ATmega328P

* **Memory :**

SRAM: 2 KB

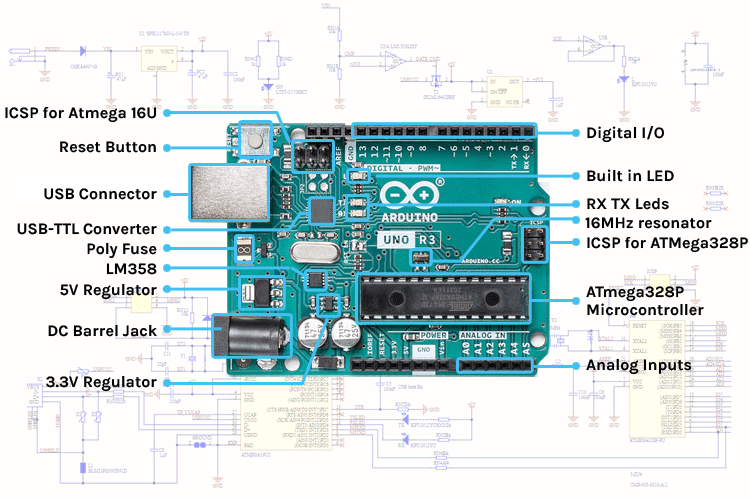
FLASH MEMORY: 32 KB (0.5 KB used by bootloader)

EEPROM: 1 KB

* **I/O Pins :**

Digital I/O pins: 14 (of which 6 can be used as PWM outputs)

Analog input pins: 6



***Assignment No.:2 Date: 9-8-2024***

**Q1. *What is an Encoding format List down encoding formats for various types of data  ( Text, Number, Photo, Audio, Video).***

**A1.** Encoding is like changing the way data is written so that it can be easily stored, sent, or processed by a computer. It’s like translating information into a language that the computer can understand.

**Encoding Formats for Different Data Types**

**Text Encoding**

* ASCII: American Standard Code for Information Interchange, used for English characters.
* UTF-8: Unicode Transformation Format-8, most commonly used for representing text in websites and applications.
* UTF-16: Unicode Transformation Format-16, used for languages with larger character sets.
* ISO-8859-1: Latin-1, used for Western European languages.

Real-World Applications of Text Encoding

* Web pages
* Email

**Number Encoding**

* Integer: Stores whole numbers (e.g., int, long, short).
* Floating-point: Stores real numbers with decimal points (e.g., float, double).
* Fixed-point: Represents fractional numbers with a fixed number of decimal places.
* Binary Coded Decimal (BCD): Represents decimal digits using binary code.

Applications of Number Encoding

* Scientific calculations
* Image processing

Image Encoding

* JPEG: Joint Photographic Experts Group, lossy compression for photographs.
* PNG: Portable Network Graphics, lossless compression for images with transparent backgrounds.
* GIF: Graphics Interchange Format, supports animation and transparency, but often limited color palette.
* BMP: Bitmap, uncompressed image format.
* TIFF: Tagged Image File Format, supports various image types and compression methods.

Real-World Applications of Image Encoding

* Digital photography
* Web graphics

**Audio Encoding**

* **MP3:** MPEG-1 Audio Layer 3, lossy compression for music.
* **AAC:** Advanced Audio Coding, often used for higher quality audio.
* **WAV:** Waveform Audio File Format, uncompressed audio format.
* **FLAC:** Free Lossless Audio Codec, lossless compression for audio.

**Real-World Applications of Audio Encoding**

* **Music streaming**
* **Voice communication**

Video Encoding

* H.264/AVC: Advanced Video Coding, widely used for video compression.
* HEVC/H.265: High-Efficiency Video Coding, newer standard offering better compression.
* VP9: Open-source video codec developed by Google.
* AV1: Royalty-free video codec with improved compression efficiency.

Real-World Applications of Video Encoding

* Online video streaming
* Video editing

***Assignment No.:3 Date: 9-8-2024***

***Q1***. Explain Basic Structure of an Arduino Program.

A1An Arduino program, also known as a sketch, is where the magic happens when bringing your electronics projects to life. It’s like writing a script that tells your Arduino board exactly what to do!

Basic Structure:

Every Arduino sketch is built around two main functions:

1. setup() function
   * When it runs: Just once, when your Arduino starts up or resets.
   * What it does: This is where you set the stage, initializing your hardware components, deciding which pins will be inputs or outputs, and setting any initial values.
2. loop() function
   * When it runs: Over and over again, endlessly, after setup() is done.
   * What it does: Here’s where the real action is! The loop contains the main logic of your program, telling the Arduino what to do and how to interact with the world, continuously.

Key Points:

* Both functions are wrapped in curly braces {} to keep things organized.
* Each instruction inside these functions ends with a semicolon ; to signal the end of a command.

Example Sketch:

Let’s take a look at a simple yet classic sketch that makes an LED blink on and off every second. It’s like making your Arduino say “hello” to the world!

cpp

Copy code

void setup() {

pinMode(13, OUTPUT); // Set pin 13 as an output

}

void loop() {

digitalWrite(13, HIGH); // Turn the LED on

delay(1000); // Wait for 1 second

digitalWrite(13, LOW); // Turn the LED off

delay(1000); // Wait for 1 second

}

* What it does: This sketch commands the Arduino to turn on an LED connected to pin 13, wait for a second, turn it off, and wait another second. Then it repeats, creating that familiar blink-blink rhythm. It's a simple yet powerful demonstration of how an Arduino can control the physical world around it!