VISHVESHVAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELGAVI - 590098

An internship training report on

"VEGA Processor & its applications"
in Association
with C-DAC and IETE Students Forum

Submitted in the partial fulfillment in the requirement for the award of Degree of
Bachelor of Engineering in Electronics and Communication Engineering

Submitted by

H BHARATH BHAT (1JS21EC052)

HRUSHIK RAJ S (1JS21EC058)

KARTHIK V KRISHNAN (1JS21EC168)

PRAJWAL S (1JS21EC103)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

(Accredited by NBA for the Academic Years 2017-18, 2018-19 and 2019-20)
JSS ACADEMY OF TECHNICAL EDUCATION
(Affiliated to Visvesvaraya Technological University, Belagavi, Karnataka, INDIA) Approved by All India Council for Technical Education, New Delhi
UG Programs Accredited by NBA- ECE, CSE, ISE, Mech, Civil, IEM, E&IE
Dr. Vishnuvardhan Road, Bengaluru-560060

INTERNSHIP TRAINING CERTIFICATE

This is to certify that the Internship Training work entitled "VE-GA Processor & its applications" in Association with C-DAC and IETE Students Forum has been successfully carried out by

H BHARATH BHAT (1JS21EC052), HRUSHIK RAJ S (1JS21EC058), KARTHIK V KRISHNAN (1JS21EC064), PRAJWAL S (1JS21EC103),

a bonafide student of **JSS Academy of Technical Education** in partial fulfilment of the requirement for the award of degree of **Bachelor of Engineering** in **Electronics and Communication Engineering** of **Visvesvaraya Technological University, Belagavi** during the academic year **2023-2024**. It is certified that all corrections/suggestion indicated for Internal Assessment have been incorporated in the report. The Internship Training report has been approved and it satisfies the academic requirement in respect of Internship work for the said degree.

ACKNOWLEDGEMENT

The successful completion of any task would be incomplete without the mention of people whose constant guidance and encouragement crowned my effort with success.

I express my gratitude to, **Dr. Bhimasen Soragaon**, Principal, JSS Academy of Technical Education, and **Dr. PM Shivakumaraswamy**, Professor & HOD, Dept. of ECE, JSS Academy of Technical Education, for his valuable support and encouragement. I express my thanks to the Department of Electronics and Communication who provided me such a great internship opportunity in *VEGA Processor & its applications.*

I am are very grateful for having such a chance for being the part of this opportunity. I would also like to thank the Internship Coordinators **Dr. Thejaswini P, Mrs. Sunitha L Shirahatti** and **Mrs. Gunasagari G S** for helping us present this Internship successfully.

Finally, I take this opportunity to extend my earnest gratitude towards **C-DAC and IETE Students Forum**.

Regards,

H BHARATH BHAT (1JS21EC052),

HRUSHIK RAJ S (1JS21EC058),

KARTHIK V KRISHNAN (1JS21EC168),

PRAJWAL S (1JS21EC103),

VISION AND MISSION OF THE INSTITUTE

Vision:

To be among the finest Institutions providing Engineering and Management Education empowered with research, innovation and entrepreneurship.

Mission:

1. Strive towards Excellence in teaching–learning process and nurture personality development.
2. Encourage Research, Innovation & Entrepreneurship.
3. Train to uphold highest ethical standards in all activities.

VISION AND MISSION OF ECE DEPARTMENT

Vision:

Enable the students to achieve academic excellence at par with premier institutes and to meet the requirements of industry and research in the field of Electronics and Communication Engineering.

Mission:

1. Orient the students to meet global challenges in the field of Electronics and Communication Engineering with state-of-the-art facilities and good teaching-learning process.
2. Impart knowledge in the emerging areas through Industry Institute-interaction, seminars, workshops and conferences.
3. Provide a strong foundation in core subjects to enable the students for continuous learning and research.
4. Facilitate students for soft skills and professional ethics.

TABLE OF CONTENTS

INTRODUCTION

A microcontroller is a compact integrated circuit that contains a processor core, memory, and programmable input/output peripherals all on a single chip. These devices are designed to execute specific tasks within embedded systems, ranging from simple consumer electronics to complex industrial applications. Microcontrollers are programmed to control various functions, such as monitoring sensors, processing data, and controlling actuators. They are widely used in automation, robotics, IoT devices, and many other applications due to their low cost, low power consumption, and versatility.

About CDAC

The Centre for Development of Advanced Computing (C-DAC) stands as a pioneering Research and Development (R&D) organization under the Ministry of Electronics and Information Technology (MeitY) in India. Established in 1988, C-DAC's inception was driven by the need to develop indigenous supercomputing capabilities after the denial of supercomputer imports by the USA. Since then, C-DAC has been instrumental in the development of multiple generations of supercomputers, starting with PARAM, boasting 1 GF in 1988. Concurrently, C-DAC ventured into Indian Language Computing Solutions through its GIST group and aligned its efforts with the National Centre for Software Technology (NCST) to further work in this domain.

Additionally, C-DAC absorbed various entities such as the Electronic Research and Development Centre of India (ER & DCI) under the purview of MeitY. These entities, initially adjuncts of state electronic corporations, focused on applied electronics and technology. Over time, C-DAC's innovative ecosystem fostered the expansion into diverse areas like Health Informatics, while NCST continued its focus on Software Technologies. Education and training became integral, beginning in 1994, evolving into robust initiatives to address the Indian industry's needs, serving as finishing schools for professionals.

Today, C-DAC has evolved into a premier R&D institution in Information Technologies and Electronics (IT&E) in India. It plays a pivotal role in enhancing national technological capabilities, aligning with global advancements in the field, and addressing evolving market demands in selected foundational areas. As an institution dedicated to high-end R&D, C-DAC is deeply entrenched in the IT revolution, consistently developing capacities in emerging technologies, leveraging its expertise to devise and deploy IT solutions across various sectors of the economy. This journey is driven by C-DAC's mandate from the Ministry of Electronics and Information Technology, alongside other stakeholders, to innovate, collaborate, and contribute to India's digital transformation.

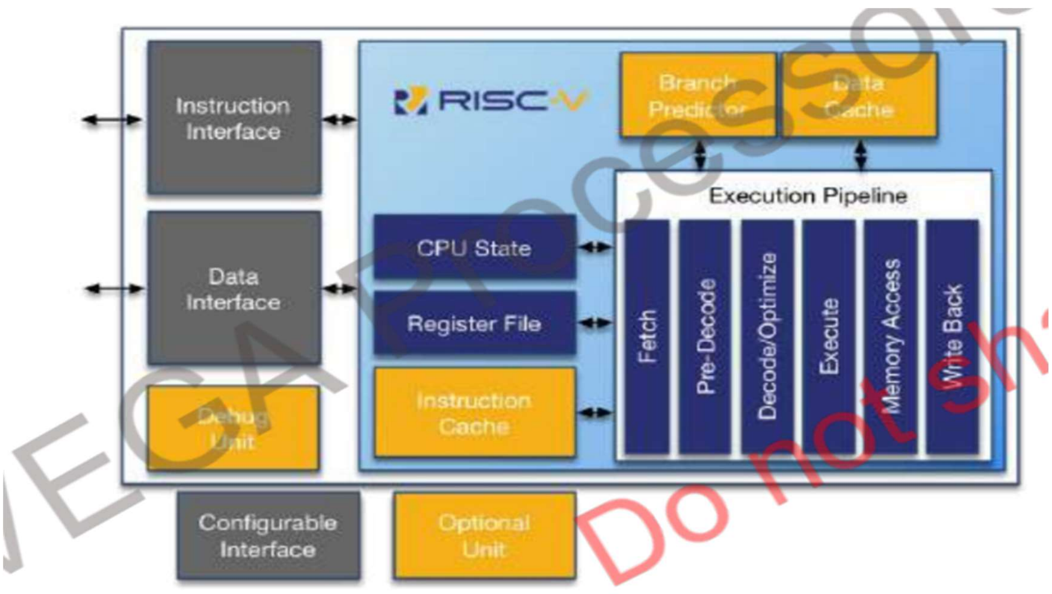
RISC-V: A Revolutionary Open-Source Instruction Set Architecture (ISA)

Introduction:

RISC-V is an open-source Instruction Set Architecture (ISA) designed to be simple, extensible, and suitable for a wide range of applications. Unlike proprietary ISAs such as ARM or x86, RISC-V is developed collaboratively by a global community of academics and industry experts. Its open nature fosters innovation, enabling anyone to design, implement, and customize RISC-V processors without the constraints of licensing fees or proprietary restrictions. This page provides an overview of RISC-V, its features, and its significance in the world of computing.

Features of RISC-V:

1. **Simplicity**: RISC-V follows the Reduced Instruction Set Computing (RISC) philosophy, favoring simplicity and efficiency. It features a small set of instructions, each with a clear and well-defined purpose. This simplicity enhances compiler optimization, reduces hardware complexity, and facilitates easier verification and testing.



2. **Modularity and Extensibility**: One of the key strengths of RISC-V is its modular design. The ISA is divided into a base integer instruction set and optional extension modules. This modular approach allows users to tailor RISC-V processors to specific application requirements by selectively adding extension modules such as floating-point arithmetic, vector processing, cryptography, or custom instructions.

3. **Scalability**: RISC-V is designed to scale from microcontrollers and embedded systems to high-performance computing platforms. It supports various instruction lengths (32-bit, 64-bit, and 128-bit) and provides flexibility in defining custom instruction formats, enabling efficient implementations across a wide range of devices and performance levels.

4. **Openness and Collaboration**: RISC-V's open nature encourages collaboration and innovation within the hardware community. Anyone can access the RISC-V specifications, contribute to its development, and implement RISC-V processors without the barriers imposed by proprietary ISAs. This openness promotes diversity in processor designs and fosters rapid innovation in the semiconductor industry.

5. **Portability and Compatibility:** RISC-V is designed to be architecture-neutral, meaning it can be implemented on various hardware platforms and seamlessly run software written for different RISC-V implementations. This portability and compatibility make RISC-V an attractive choice for both hardware developers and software engineers seeking to build scalable and interoperable systems.

RISC-V ISA:

The RISC-V ISA defines the architecture's instruction set, including its syntax, semantics, and encoding. It is organized into several base standard extensions, each targeting specific functionalities:

- RV32I/RV64I/RV128I: Base integer instruction sets for 32-bit, 64-bit, and 128-bit architectures, respectively.
- RV32M/RV64M/RV128M: Multiply extension for integer multiplication and division operations.
- RV32A/RV64A/RV128A: Atomic extension for supporting atomic memory operations.
- RV32F/RV64F/RV128F: Single-precision floating-point extension.
- RV32D/RV64D/RV128D: Double-precision floating-point extension.

- RV32C/RV64C/RV128C: Compressed extension for reducing code size.
- RV32V/RV64V/RV128V: Vector extension for SIMD (Single Instruction, Multiple Data) operations.

In addition to these base standard extensions, RISC-V also supports custom extensions, allowing users to define and incorporate proprietary instructions tailored to their specific needs.

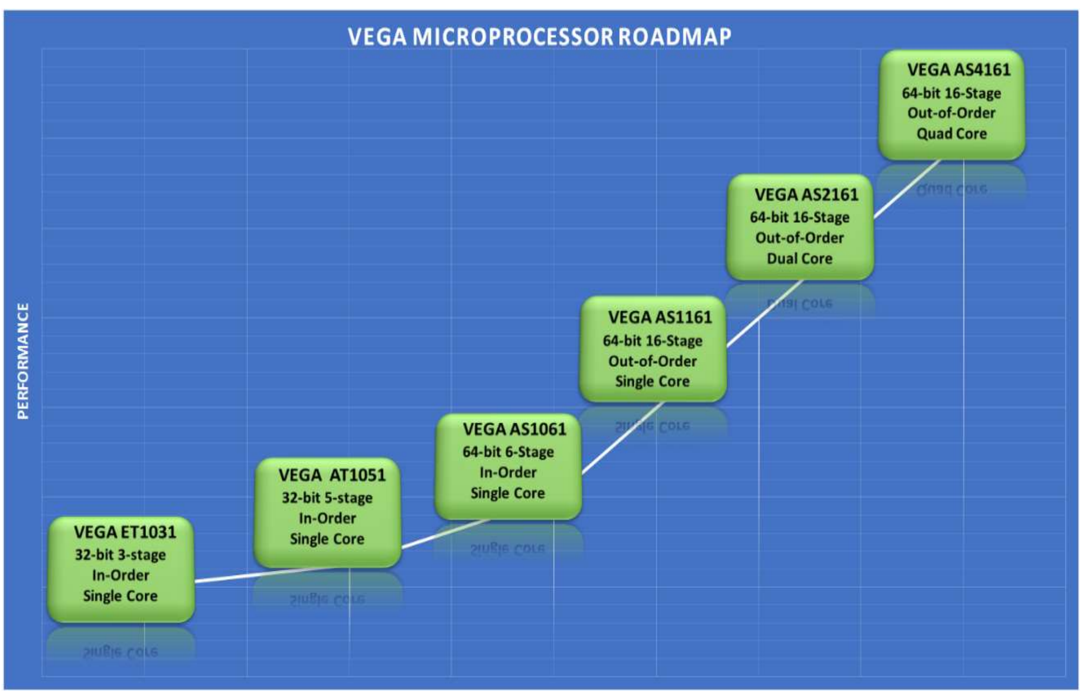
RISC-V ISA Naming Conventions:

The ISA naming convention has the format as follows:

- RV, which stands for RISC-V,
- sequence of numbers - #32, #64, or #128.
- Sequence of letters.
- Required extension -> "I" extension, which stands for integer,
- RV + word-width + extensions,
- RV32IMC: 32bit, integer, multiplication, compressed.

VEGA Processors:

As part of the DIR-V Program, Centre for Development of Advanced Computing (C-DAC) developed a series of Microprocessors under the name "VEGA Processors".



VEGA ET1031:

THEJAS32 is a SoC based on 32-bit VEGA ET1031 Microprocessor

- VEGA ET1031 - a 32-bit high performance microcontroller class processor comprising of a 3-stage in-order RISC-V based core.This processor design is compact and efficient and is targeted for applications like sensor fusion, smart meters, small IoT devices, wearable devices, electronic toys, etc.
- Can be used as an effective work horse in low power IoT applications.
- Taped out in SilTerra 130nm process.
- The peripherals available in THEJAS32 SoC are Interrupt Controller, Timers, RAM, GPIO,SPI, UART, I2C, :PWM, ADC and External interrupt capable GPIOs (GPIO0-GPIO12).

Key features

- RISC-V (RV32IM) Instruction Set Architecture
- 3-stage in-order pipeline implementation
- Harvard architecture (separate instruction and data buses)
- High-performance multiply/divide unit
- Configurable AXI4 or AHB external interface
- Low interrupt latency
- Vectored interrupt support
- 128 pin LQFP package
- Low power operation
- Upto 2 MB boot flash via SRAM
- Clock frequency upto 100 MHZ.

Applications:

- Sensor fusion
- Smart Meter
- System supervisors
- Remote sensors
- Small IoT devices and many more.

Key Features	VEGA ET1031	VEGA AT1051	VEGA AS1061	VEGA AS1161	VEGA AS2161	VEGA AS4161
RISC-V ISA	32-bit RV32IM	32-bit RV32IMAFC	64- bit RV64IMAFDC	64-bit RV64IMAFD	64-bit RV64IMAFD	64-bit RV64IMAFD
No of cores	1	1	1	1	2	4
Pipeline	In-order	In-order	In-order	Out-of-Order	Out-of-Order	Out-of-Order
Pipeline stages	3-Stage	5-Stage	6-Stage	13-16 Stage	13-16 Stage	13-16 Stage
Superscalar	No	No	No	Yes	Yes	Yes
Processor modes	Machine	Machine/ Supervisor/User	Machine/ Supervisor/User	Machine/ Supervisor/User	Machine/ Supervisor/User	Machine/ Supervisor/User
MMU	Optional	Yes	Yes	Yes	Yes	Yes
Debug	Optional	No	Yes	Yes	Yes	Yes
Branch Predictor	No	Yes	Yes	Yes	Yes	Yes
L1 ICache	TIM	8KB	8KB	32KB	32KB	32KB
L1 DCache	TIM	8KB	8KB	32KB	32KB	32KB
L2 Caches	No	No	No	No	512KB	1024KB
Bus Interface	AHB/AXI4	AHB/AXI4	AHB/AXI4	AHB/AXI4/ACE	AHB/AXI4/ACE	AHB/AXI4/ACE
IEEE 754-2008 compliant FPU	No	Single precision	Single and Double precision	Single and Double precision	Single and Double precision	Single and Double precision
Availability	Now	Now	Now	Now	Now	Now

THEJAS32 SoC ASIC:

THEJAS32 SoC is based on the VEGA ET1031 processor, which is a 32 bit single core in-order, 3-stage pipeline processor. This processor is based on the open source RISC-V Instruction Set Architecture and operates at a frequency of 100MHz. The SoC also includes peripherals like 256KB internal SRAM, Three UARTs, Four SPIs, Three TIMERs, Eight PWMs, Three I2C interfaces, 32 GPIOs etc.

UART for THEJAS32:

1. THEJAS 32 features 3 built-in UART ports for serial communication with devices like GSM, GPS, Bluetooth, etc.

2. UART port 0 (RX0 & TX0) is used for communication with a PC/laptop via USB, while UART port 1 (RX1 & TX1) and UART port 2 (RX2 & TX2) are utilized for communication with serial devices.

3. Serial communication occurs at TTL logic levels (3.3V for 1, 0V for 0) and supports various baud rates such as 4800, 9600, 19200, etc., with a default baud rate of 115200 for UART-0.

I2C for THEJAS32:

1. THEJAS 32 offers 3 built-in I2C ports for serial communication.

2. I2C port 0 (SCL0 & SDA0) and I2C port 1 (SCL1 & SDA1) are used for communication with external devices, while I2C port 2 (SCL2 & SDA2) is internally utilized for ADC communication.

3. Data transmission occurs over the SDA line with clock pulses on the SCL line, and data is shifted out with MSB first, expecting an acknowledge after each byte in transmit/receive mode.

SPI for THEJAS32:

1. THEJAS 32 includes 4 built-in SPI ports for serial communication.

2. SPI port 0, SPI port 1, and SPI port 2 are utilized for communication with external devices, while SPI port 3 is reserved for boot flash interface.

3. The master device generates clock and chip select signals, configuring clock frequency, polarity, and phase with respect to data exchange during SPI communication.

ADC for THEJAS32:

1. THEJAS 32 features a 4-channel ADC for analog communication via pins A0 - A3.

2. ADC communication is facilitated through I2C port 3, connected to an ADS1015 for analog channel measurements.

3. The master (microcontroller) provides clock signals on SCL pin, and data transfer occurs via SDA pin, with the ADS1015 not driving the SCL pin and only accepting positive voltages.

GPIO for THEJAS32:

1. THEJAS32 integrates 32 in-built GPIO pins for digital communication, initially set as inputs upon system reset.

2. Reading the data register provides the current status of GPIO pins, while writing to it only affects pins configured as outputs.

3. Bit masking is supported in both read and write operations through address lines, with the GPIO data direction register configuring pins as inputs or outputs based on bit values.

PWM for THEJAS32:

1. THEJAS32 offers 8 PWM channels for communication with PWM devices via PWM pins PWM0 - PWM8.

2. PWM controller operates in three modes: Idle, One short, and Continuous, allowing flexible PWM signal generation.

3. Modes include Idle mode, where PWM output remains static; One short mode, generating PWM signals for a defined number of cycles; and Continuous mode, producing PWM signals continuously.

Timers for THEJAS32:

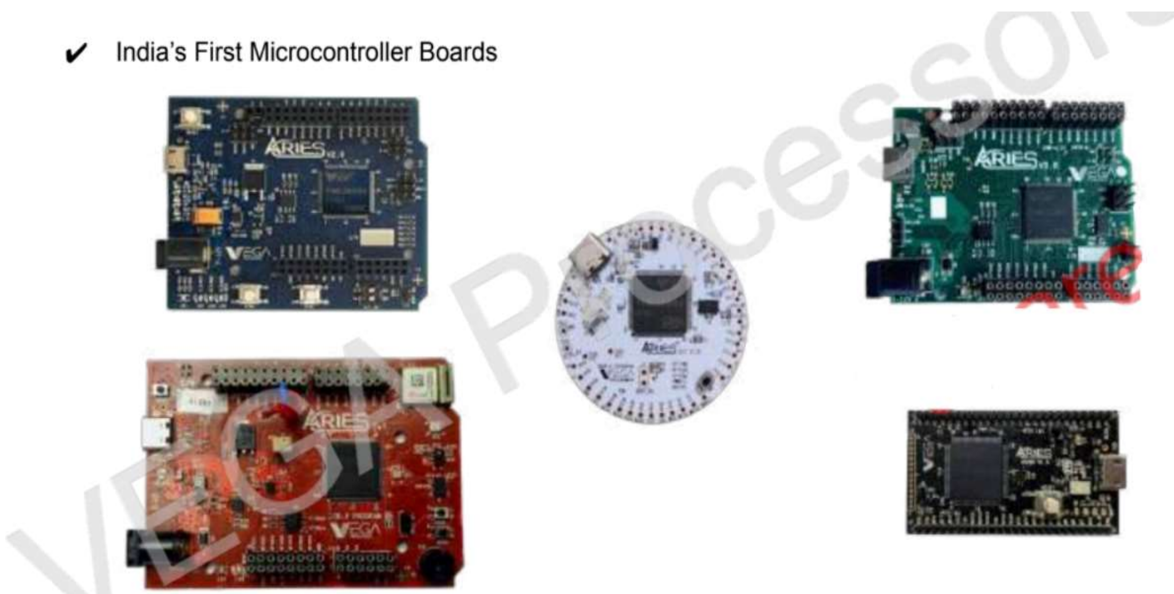
1. THEJAS32 microcontrollers feature three 32-bit timers (timer0, timer1, and timer2) with decrementing counters on each clock tick.

2. Timers support two operation modes: Free Running Mode and User Defined Count Mode.

3. In Free Running Mode, the timer reloads upon reaching 0, generating an interrupt and continuing to decrement until a reset or disable event. In User Defined Count Mode, timers generate interrupts upon reaching a programmed value, reloading with a user-defined value for continued operation.

Applications of THEJAS32 SOC:

- Sensor fusion
- Smart Meter
- System supervisors
- Remote sensors
- Small IoT devices
- Wearable devices
- Toy and electronic education equipment
- Legacy 8/16-bit applications
- Industrial networking and many more...



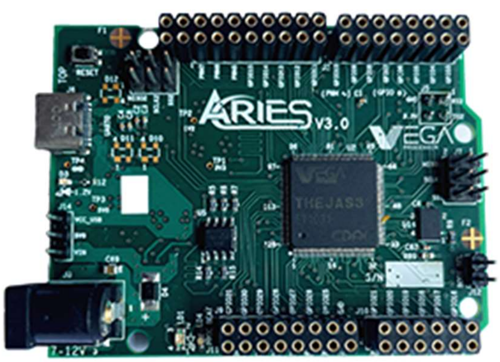
Tech Fusion Cruiser

Introduction

This project involves the creation of a Bluetooth-controlled car, which demonstrates the integration of various electronic components and modules to achieve wireless control. The car is designed to be operated via Bluetooth, leveraging the HC-05 Bluetooth module for wireless communication. The core of the project is the Aries development board, which is used to execute the control code and manage the car's operations.

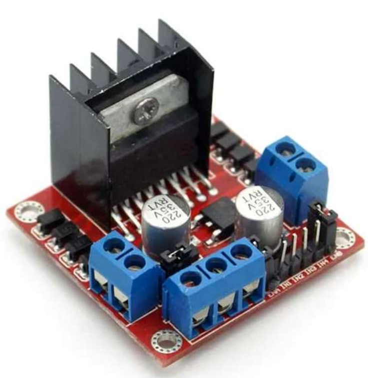
For motor control, the L298N motor driver is employed, enabling precise control of the four DC motors that power the car's movement. The entire setup is powered by a portable power bank, providing the necessary electrical power to all components. This project exemplifies a practical application of Bluetooth technology in remote vehicle control, making use of microcontroller programming, motor control, and wireless communication.

Components Required:



1. Aries V3.0 Development Board

2. HC-05 Wireless Bluetooth RF Transceiver

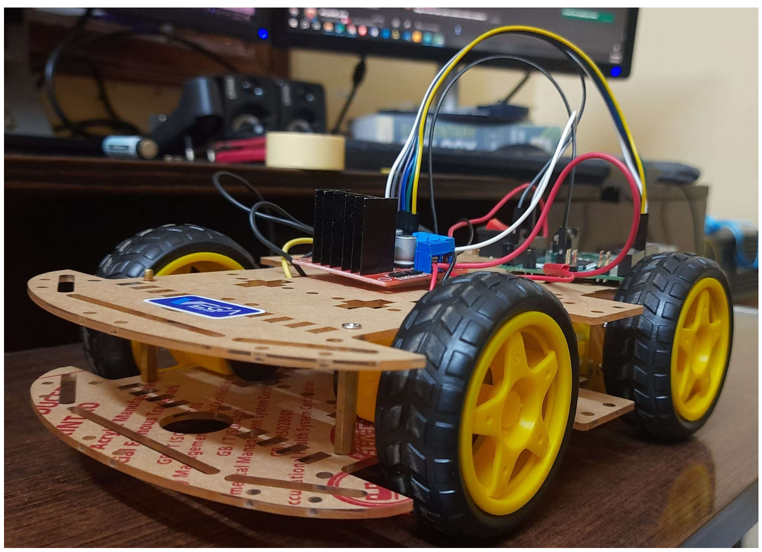


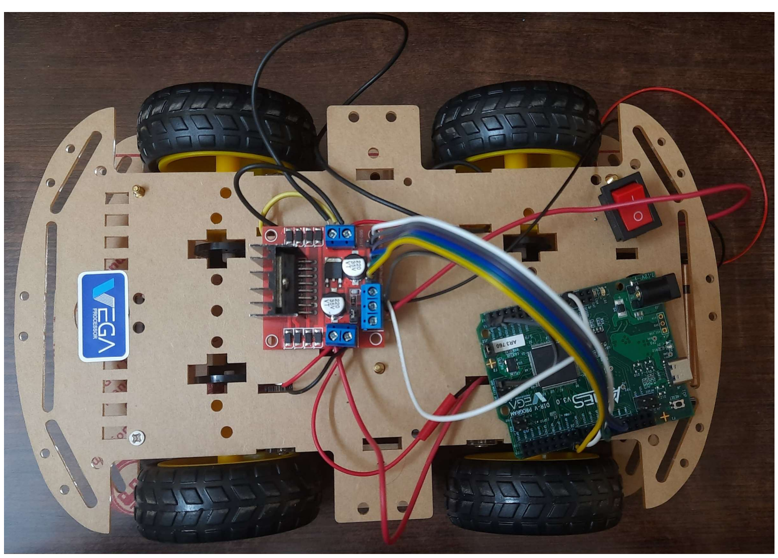
3. L298N Motor Drive Controller Board Module Dual H Bridge

4. 4-wheel Drive Robot Smart Car

5. 10,000mAh Power Bank/ Batteries

Assembled Model:





Connections:

HC05 and Aries Board	
HC05	**Aries Board**
12V & 5V	5V
GND	GND
RXD	TX1
TXD	RX1

L298N Motor Driver and Aries Board	
L298N	**Aries Board**
VCC	5V
5V In/Out	VIN
GND	GND
ENA	PWM 1
IN1	GPIO 13
IN2	GPIO 15
IN3	GPIO 14
IN4	GPIO 16
ENB	PWM2

L298N Motor Driver and Motors	
L298N	**Motors**
Motor Terminal 01	Front and Rear Left Motors
Motor Terminal 02	Front and Rear Left Motors
Motor Terminal 03	Front and Rear Right Motors
Motor Terminal 04	Front and Rear Right Motors

Code:

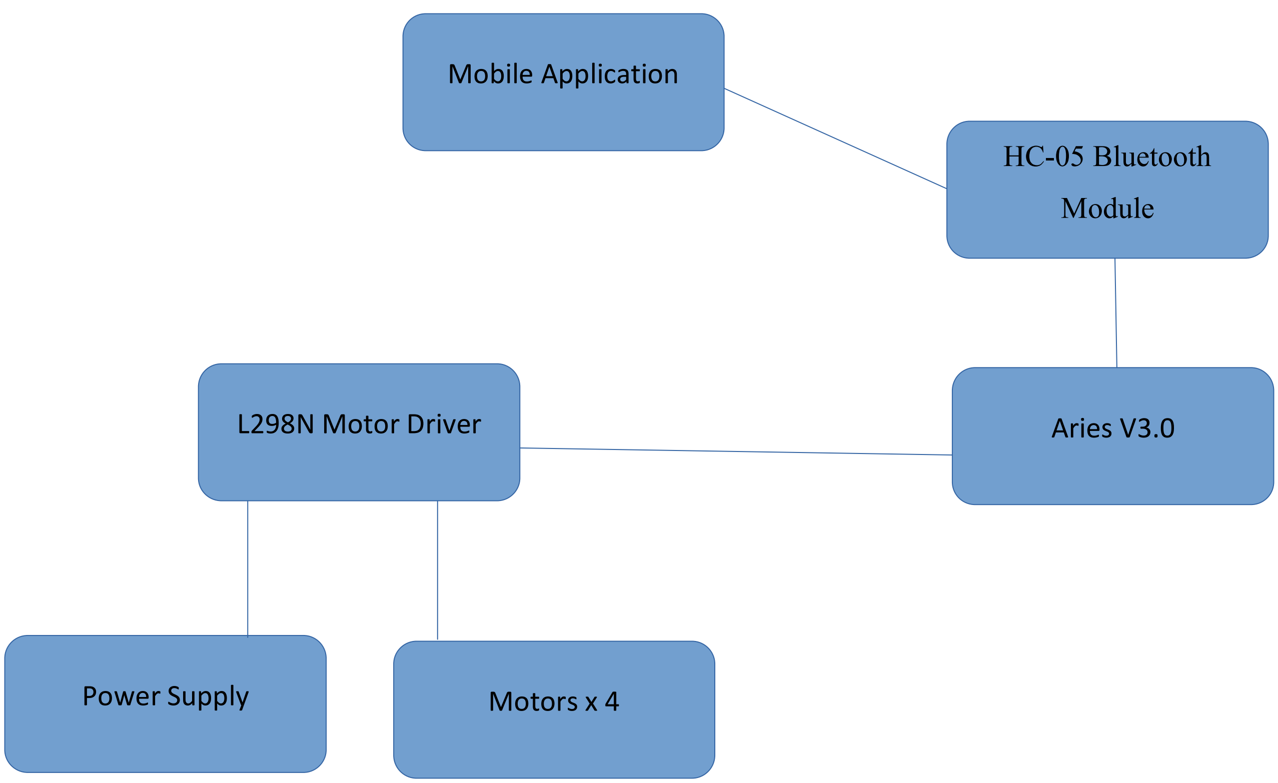
```
#include<pwm.h>
#include<esp8266.h>

ESP8266Class Bluetooth(1);
//UARTClass mySerial(1);

char receiveBuffer[4096] = {0, };
int count;

//Motor Connections
//Change this if you wish to use another diagram
#define EnA 1       //Connect Enable pin to PWM 1
#define In1 13      //Connect input pin to GPIO-13
#define In2 15      //Connect input pin to GPIO-15
#define EnB 2       //Connect Enable pin to PWM 2
#define In3 14      //Connect input pin to GPIO-14
#define In4 16      //Connect input pin to GPIO-16
char in=0;


void setup(){
  pinMode(EnA, OUTPUT);
  pinMode(In1, OUTPUT);
  pinMode(In2, OUTPUT);
  pinMode(EnB, OUTPUT);
  pinMode(In3, OUTPUT);
  pinMode(In4, OUTPUT);
  PWM.PWMC_Set_Period(EnA, 800000);
  PWM.PWMC_Set_Period(EnB, 800000);
  Serial.begin(115200);  // UART-0
  Bluetooth.begin(9600); // UART-1
  delay(1000);
```

```
}

void forward() {
  analogWrite(EnA, 800000);
  digitalWrite(In1,HIGH);
  digitalWrite(In2,LOW);
  analogWrite(EnB, 800000);
  digitalWrite(In4,HIGH);
  digitalWrite(In3,LOW);
}

void left(){
  analogWrite(EnA, 800000);
  digitalWrite(In1,LOW);
  digitalWrite(In2,HIGH);
  analogWrite(EnB, 800000);
  digitalWrite(In4,HIGH);
  digitalWrite(In3,LOW);
}

void right(){
  analogWrite(EnA, 800000);
  digitalWrite(In1, HIGH);
  digitalWrite(In2,LOW);
  analogWrite(EnB, 800000);
  digitalWrite(In4,LOW);
  digitalWrite(In3,HIGH);
}

void reverse(){
  analogWrite(EnA, 800000);
  digitalWrite(In1,LOW);
  digitalWrite(In2,HIGH);
  analogWrite(EnB, 800000);
```

```
  digitalWrite(In4,LOW);
  digitalWrite(In3,HIGH);
}

void brake() {
  digitalWrite(In1, LOW);
  digitalWrite(In2, LOW);
  digitalWrite(In3, LOW);
  digitalWrite(In4, LOW);
}

void loop(){
  count = 0;
  for(;count<=0;count++){
    receiveBuffer[count] = 0;
  }
  Serial.print("Ready");
  while(1){
    count = 0;
    while(Bluetooth.available() > 0){
      receiveBuffer[count++] = Bluetooth.read();
    }
    if(receiveBuffer[count-1] == '\n'){
      receiveBuffer[count-1] = '\0';
      break;
    }
  }

  Serial.println(receiveBuffer);

  if(strcmp(receiveBuffer, "1")==0){
    forward();
    delay(100);
  }
```

```
else if(strcmp(receiveBuffer, "2")==0){
  left();
}
else if(strcmp(receiveBuffer, "4")==0){
  right();
}
else if(strcmp(receiveBuffer, "3")==0){
  reverse();
}
else if(strcmp(receiveBuffer, "0")==0){
  brake();
}
else{
  brake();
}
}
```

Code Link: https://github.com/bhatbharath/cdac_vega_tech_fusion_cruiser/



How It Works:

As mentioned, the components used are the Aries V3.0 development board, an HC-05 wireless Bluetooth RF transceiver, an L298N Motor Drive Controller Board Module Dual H Bridge, a 4-wheel drive robot smart car and 10,000mAh batteries or a power bank. In this section, we will explain how all these components work together seamlessly to power the tech fusion cruiser.

The 4-wheel drive robot smart car forms the basis of the cruiser on which the rest of the components have been built or programmed. The L298N Motor Drive Controller is responsible for powering the cruiser with the help of the 10,000mAh power bank or the same capacity in terms of batteries.

On this physical set-up, operates the HC-05 Bluetooth sensor, powered by the Aries v3.0 development board. We programmed the board to accept commands from a specified control device, hence giving us remote control over the motion of the cruiser via the Bluetooth sensor. Once programmed and physically set-up, the cruiser is entirely responsive to all commands from the control device, within its range of motion.

The cruiser receives commands via the Bluetooth sensor, which is then passed on to the board. From here, through the L298N motor driver, the board directly executes the given commands, causing the cruiser to respond as directed by the control device.

Through rigorous testing, we have demonstrated that the cruiser reliably responds to remote commands, confirming the effectiveness of our integration and programming efforts. The project highlights the potential of combining readily available electronic components to create innovative and practical solutions.

Applications:

One of the most prominent characteristics of the tech fusion cruiser is its ability to be operated from a distance, which opens up a whole avenue of possible applications:

- It can be used to move small amounts of emergency medical supplies in areas where it is not practical to carry anything on one's body.
- It can similarly be used to send medical supplies into areas where human presence is unwanted or where humans cannot reach.

- It can be used to conduct surface-level surveillance in remote areas where humans cannot reach.
- It can be used for military operations in situations where using soldiers would be life-threatening.
- It can be used to capture footage of areas where humans cannot reach, for research purposes.
- It can be modified to bring back samples from unreachable areas similarly for research purposes.
- It can be used to carry instructions to places where humans cannot reach easily.
- It can be fitted with multiple peripherals to serve various objectives, as per the need of the situation.
- It can be used to conduct surveying of large farmlands without the need for human effort.
- It can be similarly used for geographical studies where it would be more convenient than sending multiple teams of humans to scope out the area.
- It can also be used for geographical studies by sending into places that humans cannot access very easily, such as volcanic craters.

Conclusion:

The Tech Fusion Cruiser exemplifies the successful fusion of technology and innovation. It serves as a testament to the power of modern electronics and programming in creating dynamic and responsive systems. This project not only meets its intended objectives but also lays the groundwork for future enhancements and applications in the realm of remote-controlled robotics.

References:

- https://vegaprocessors.in/