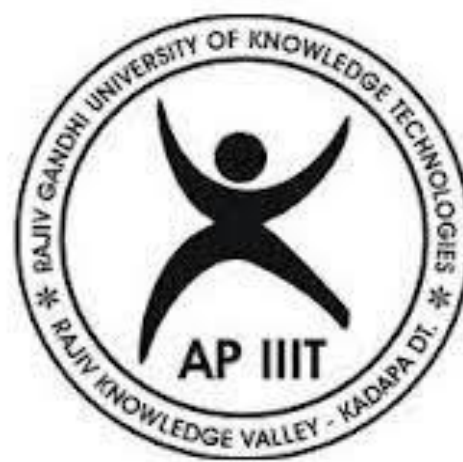
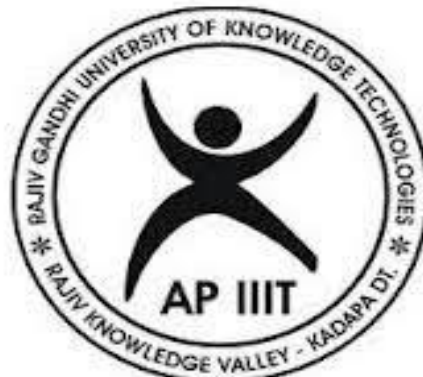


A Project Report
on
EMBEDDED SOFTWARE TECHNOLOGIES
submitted by
Kenchakkagari Prathyusha (R170902)
in partial fulfillment of the requirements for the degree of
BACHELORS OF TECHNOLOGY IN COMPUTER SCIENCE AND TECHNOLOGY
under the guidance of
MS. P. Udaya Sree
Assistant professor
Department of computer science and engineering



Rajiv Gandhi University of Knowledge And Technologies,
RK Valley, Idupulapaya,
Kadapa (Dist), Andhra pradesh, 516330

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE AND TECHNOLOGIES



RGUKT, RK Valley

Department Of Computer Science and Engineering

CERTIFICATE

This is certify that the project entitled “**Embedded Software Technologies**” submitted by the **Kenchakkagari Prathyusha** (R170902) under our guidance and supervision for the partial fulfilment for the degree Bachelor of Technology in Computer Science and Engineering during the academic year 2022-2023 at RGUKT, RK VALLEY. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any University or Institute for the award of any degree or diploma

Mr.Satyanandhram

Head of Department,

Assistant Professor,

Computer Science and Engineering,

RGUKT, R.K Valley.

Ms.P.Udaya Sree,

Project Coordinator,

Assistant Professor,

Computer Science and Engineering,

RGUKT, R.K Valley.

DECLARATION

We hereby declare that the report of the B.Tech Project entitled with the “**Embedded Software Technologies**” which is being submitted to Rajiv Gandhi University of Knowledge Technologies, RK Valley, in partial fulfilment of the requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a bonafide report of the work carried out by me. The material contained in this report has not been submitted to any university or institution for award of any degree

Date : 19-09-2022,

Place: RGUKT, R K VALLEY.

Kenchakkagari Prathyusha,

(R170902)

Acknowledgment

The success and outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and we would not forget to thank them.

We are grateful to **Ms.P.Udaya Sree** for encouragement and more over timely support and guidance.

At the outset, I would like to thank Rajiv Gandhi University of Knowledge and Technologies, R.K Valley for providing all the necessary resources for the successful completion of my course work.

With Sincere Regards,

K.Prathyusha (R170902)

INDEX

1. ABSTRACT6

2. INTRODUCTION.....7

3. Technologies Learned8

4. Why we use C, C++ and Advance C?9

5. BITFIELDS:11

6. Advantages of Macros:18

7. CONCLUSION.....19

8. REFERENCES.....20

ABSTRACT

The aim of this project report is to give a brief idea on how Actevia and most other Embedded companies work on C **languages**, provides optimized machine instructions for the given input, which increases the performance of the embedded **system**. **It** is an Automobile Company which uses embedded **software**. **Here, we** Test codes and **debugg** it using **C, C++**.

INTRODUCTION

Engineers use C programming for embedded systems to produce efficient and compact code. C is important in embedded systems because there is often limited memory and processing power.

C programming produces very reliable code enabling embedded systems to execute mission-critical tasks.

Although not originally designed for embedded software development, **the C language allows a range of programming styles from high-level application code down to direct low-level manipulation of hardware registers.** As a result, C has become the most popular programming language for embedded systems today.

C++ is the mother language of embedded systems

So, a large number of ECUs run the transmission control, infotainment, engine management, passive and active safety systems (ABS and ADAS), navigation, security, and climate.

Although many computer languages are used for writing computer applications, the computer programming language.

Technologies Learned

We used C, Advance C and C++.



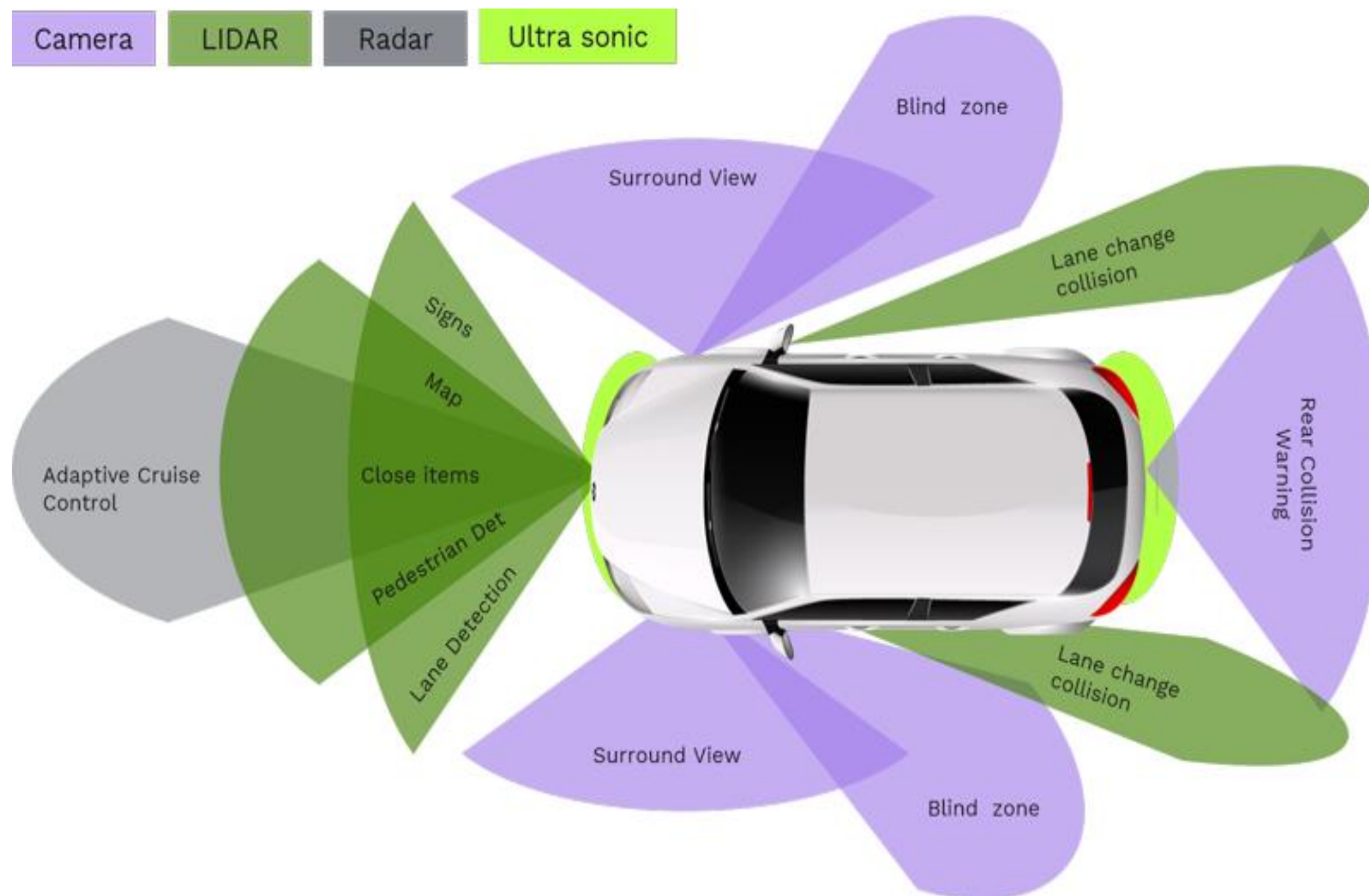
Why we use C, C++ and Advance c?

Designed to take your basic C skills to the next level and help you obtain mastery of the language by helping you understand advanced concepts of the C programming language, enabling you to master the art of problem-solving in programming using efficient, proven methods.

Engine control units (ECUs) in the embedded system are best controlled and programmed in C++ because the code can run high-level structured programming on low-level mechanisms.

Smart cars have at least 100 ECUs that fulfill versatile functions—from basic ones (engine ignition and power steering control) to comfort, access, and security.

So, a large number of ECUs run the transmission control, infotainment, engine management, passive and active safety systems (ABS and ADAS), navigation, security, and climate.



Some Advanced topics in C

These are the Advanced C topics: -

Pointers

BitFields

Memory / Dynamic Memory.

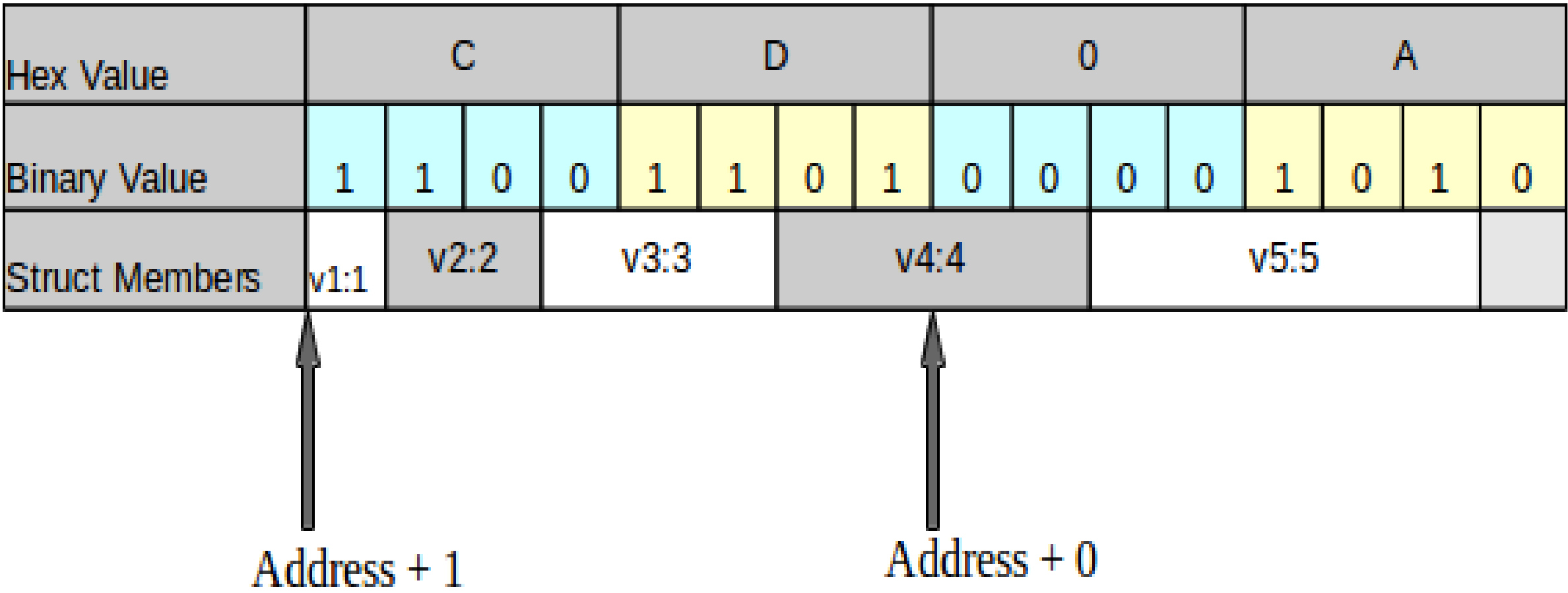
I/O & Function Topics.

Debugging / Exception Handling.

Some examples regarding Advance c:

BITFIELDS:

Use of bit fields is one of the key optimization methods in embedded C programming, because these allow one to pack together several related entities, where each set of bits and single bits can be addressed.



Example for BitFields :

```
#include<stdio.h>

struct date {
    unsigned int d:5;
    unsigned int m:4;
    unsigned int y:23;
};

int main()
{printf("Size of date is %d is %d bytes\n",sizeof(struct
date);
    struct date dt={09,06,2020};
    printf("Date is %d%d%d",dt.d,dt.m,dt.y);
}
```

output:Size of date is 4 bytes

Date is 09/06/2020

Purpose of this code is ...Here we are assiging particular required bits to the date,month,minutes.

By this we can save memory.

Interesting facts about bit fields in C

- A Special unnamed bit field of size 0 is used to force alignment on next boundary.
- We Cannot have pointers to bit field members as they may not start at a byte boundary.
- Assigning an out-of-range value to a bit field member is implementation defined (complier dependent).
- In C++, we can have static members in a structure/class, but bit fields cannot be static.

Preprocessors in C

- Preprocessor offers several features called preprocessor directives.
- Each preprocessor directive starts with a#
- The directives can be placed anywhere in the program, generally are written at the beginning.

- Preprocessor directives are:

1. Macro expansion
2. File Inclusion
3. Comments removal
4. Conditional compilation

```
#define PI 3.142
int main()
{
    int rad;
    float area;
    area =PI * rad * rad;
}
```

```
#define SIZE 5
int main(){
    int arr[SIZE];
    int i;
    for( i = 0; i < SIZE; i++ )
        printf("i=%d\n", i);
}
```

- The Statements, #define PI 3.142, #define SIZE 5 are called macro definitions or simply macro.

- During preprocessing, the preprocessor replaces every occurrence of PI with 3.142 and SIZE with 5.

A macro can be used to define the operators as

```
#define AND &&
#define OR ||
main ()
{
int f=1, x=4, y=90;
if((f<5) AND (x <= 20 OR y <=45) )
printf ("It's one-use case of macro");
else printf ("Macro is just useful");
}
```

Macro can be used to replace a condition

```
#define RANGE (a > 25 && a < 50)
main ()
{
int a = 30;
if RANGE
printf ("Within the Range");
else
printf (" Out of Range");
}
```

Macro to replace an entire C statement

```
#define FOUND printf("Welcome to CBS");  
main()  
{  
char ch;  
scanf("%c", &ch);  
if( ch == 'Y')  
FOUND  
else  
printf(" You are a trespasser ");  
}
```


Macro with arguments

```
#define AREA(x) (3.142*x*x)
main()
{
float r1 = 2.5, r2 = 3.6, result;
result = AREA(r1); // (3.142 * 2.5 * 2.5)
printf("The area with r1 is %f", result);
result = AREA(r2); // result = 3.142 * r2 * r2
printf("The area with r2 is %f", result);
}
```

```
#define ADD(x, y ,z) (z = x + y)
main()
{
int a= 10, b=20, sum;
ADD(a, b, sum); // sum= a+ b;
printf("%d", sum);
}
```

ADVANTAGES OF MACROS:

- Speed versus size the main benefit of using macros is **faster execution time**. During preprocessing, a macro is expanded (replaced by its definition) inline each time it is used. A function definition occurs only once regardless of how many times it is called.
- And while we are writing 1000 lines of code, if we modify any value. We have to check the whole program and we must modify it, but by using a macro it will modify anywhere in the program we don't want to check or it just modifies the macro declaration.
- Reduce time.

CONCLUSION

As in the learning process of “Project Report” we have studied about C and Advance C. We did lot of Programs regarding good enhancement of programming in our work. In Future also, Actevia private limited do excellence with knowledge of advance C.C provides optimized machine instructions for the given input, which increases the performance of the embedded system. Most of the high-level languages rely on libraries, hence they require more memory which is a major challenge in embedded systems.

REFERENCES

<https://www.geeksforgeeks.org/>

<https://www.tutorialspoint.org/>