

23/05/2023

Batch - 9

KUMAR, V.B
20220860

SECTION - I

CAN and UDS

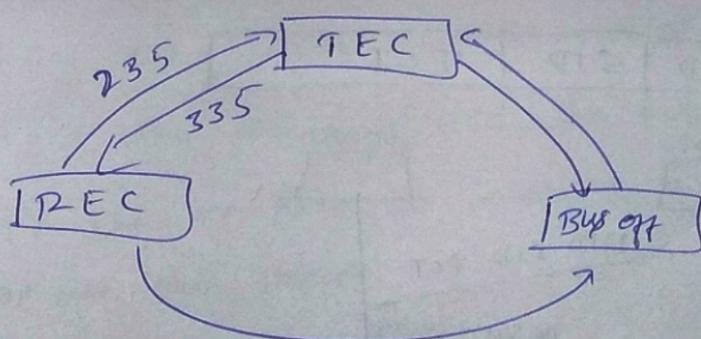
- 2) TEC (Transmit Error Counter) and REC (Receiver Error Counter).

Transmit Error Counter

When the message is transmitting the ~~is~~ less than 335 value it transmitting the It is called Transmit Error counter

Receiver Error Counter

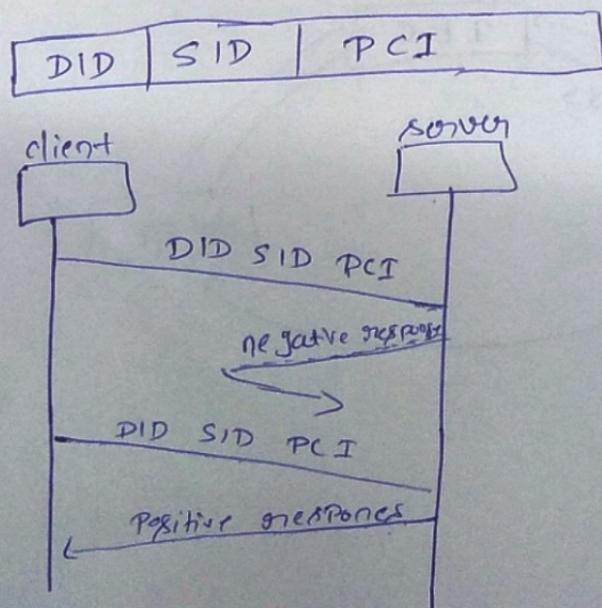
When the message is receiving the above 335 value is called Receiver error counter



①

③ Categories of services offered by UDS protocol
Stack.

- 1) Diagnostic session control.
 - 2) Diagnostic transmission.
 - 3) Data stored.
 - 4) Input and output.
 - 5) Remote activation.
 - 6) Upward and downward.
- 1) One ECU will go to bus off state. another 3 ECUs will be sending and receiving the message's according to ~~other~~ & Data.
- 5) Periodic message types of UDS?



(4)

Overload frame :- It is occurred during delay between transmitter and receiver.

Inter frame space :- It is used to wait. Dominant bit is standard frame, and receive bit form extended frame format.

SECTION - II

Embedded System & Embedded C

a>

```
#include <stdio.h>
```

```

int main()
{
    int *p = NULL;
    return 0;
}
```

The using like a NULL pointer whenever we do avoid the unwanted the errors, because we have to initialize the null to use the null pointer.

(3)

6) Interrupt Latency

For Example:- I have to watching movie in mobile suddenly one ad will come suddenly interrupt so this interrupt latency happen.

7) Barrel Shifter :- It is most significant part in ARM data flow control.

It is the center of this model. It is receiving, receiving and transmitting the and connecting connecting to the ALU and. Incrementation, and op code and Registers.

7) Concatenation operators

Whenever we need to combine of two numbers or two data where we do use the ~~operator~~ is called concatenation operator.

```
#include <stdio.h>
#include <LPC2148.h>
int main()
{
    int a, b;
    a = 10;
    b = 20;
    printf("%d %d", cat(a, b));
```

} using the concatenation function we have to combine the two no numbers.

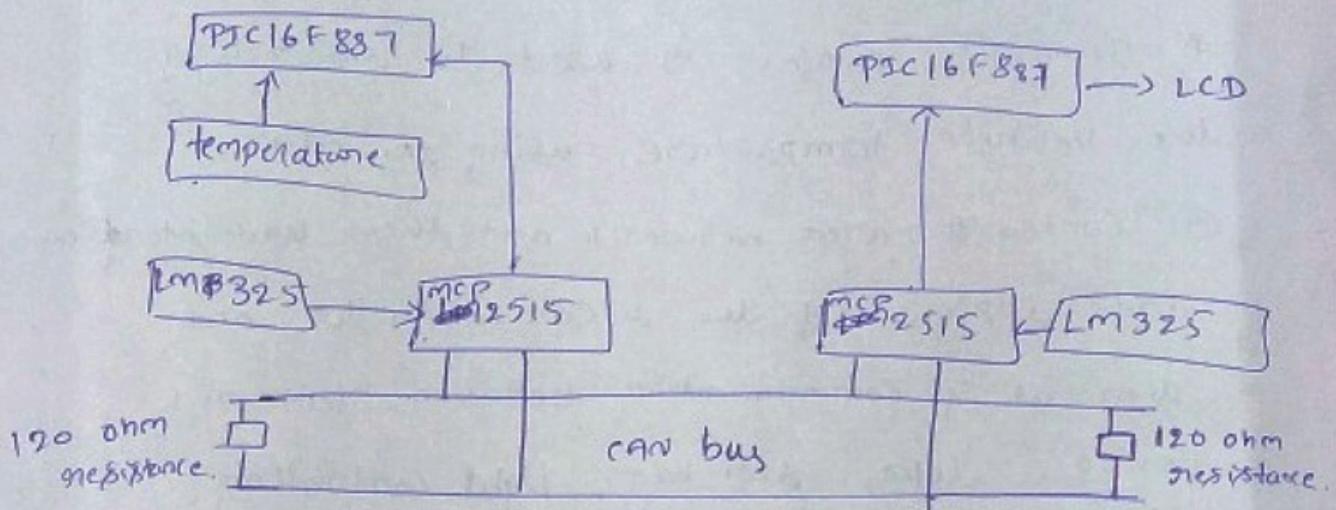
SECTION- IV

(15)

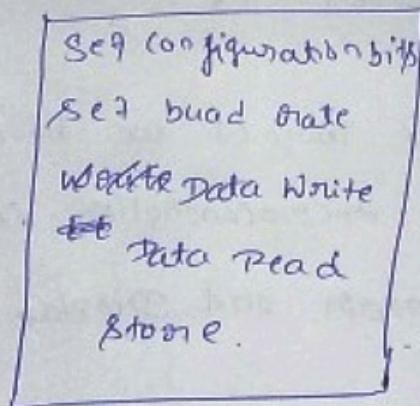
Project summaryProject nameVehicle Temperature monitoring using CAN controller

- * In this Project is used to monitoring the vehicle temperature. using the CAN Controller area network and then understanding and insight of the microcontroller and then it is communication between electronic device. like, air bag, light controlling, AC, seat belt, Brake monitoring, cruise control, engine control etc.
- * So here In this Project we have to use the PIC16F887 microcontroller, CAN LM325, and temperature sensor. and Display to use the temperature,

Block Diagram



Functional block diagram



- In this project we have to use the temperature sensor to find out the temperature in engine.
- In this we have to use the first we have set the configuration bits.

- * And then we have to set the configuration bits baud rate.
- * and then the temperature sensor LM325 is using CAN controller MCP2515 using this to find out the temperature is there how much it is there. using microcontroller PIC16F877.
- * After words it is sending back CAN using the CAN Bus. In this CAN Bus connecting with 120 ohm resistance because of impedance so we have to connect the 120 ohm.
- * From Fig 81 we have set the engine temp like upto 110°C and the temp sensor is find out the if above 110°C or below 110°C it is noted.
- * Using the display we have to display the how much the temperature is there it is display.

SECTION - 3
C - programming

(11) #include <stdio.h>
int main()
{
 int n = 9;
 if (int i, j;
 if (i == 0 || i == 1 || i == 4 || i == 5 || i == 6 || i == 7 || i == 8)
 {
 printf("*");
 }
 if (i == 2 || i == 3 || i == 8 || i == 9 || i == 10 || i == 11 || i == 12)
 {
 printf("*");
 }
 if (i == 0 || i == 1 || i == 2 || i == 3 || i == 4 || i == 5 || i == 6 || i == 7 || i == 8)
 {
 printf("*");
 }
 }
}

SECTION-IIIC-Programming

13) #include <stdio.h>

```
int main()
{
```

```
    int n, pos;

```

```
    printf("Enter number:");

```

```
    scanf("%d", &n);

```

```
    printf("Enter the position:");

```

```
    scanf("%d", &pos);

```

```
    n = n ~ (1 << pos);

```

```
    printf("n=%d", n);
}
```

12) #include <stdio.h>

#include <stdlib.h>

void point (const char s[30])

{

```
for(i=0; s[i]; i++)

```

```
    for(j=0; s[i]; j++)

```

```
        if(s[i] == s[j])

```

}

Void main () {

```
    char s[30];

```

```
    printf("Enter the string in:");

```

```
    scanf("%s", s);

```

```
    static int i=0;

```

```
    if (i<10) {

```

```
        point (s[30]);

```

```
        i++;

```

}

(a)

10) #include <stdio.h>

```
void main() {
    int a[5] = {1, 2, 3, 4, 5};
    int b[5] = {6, 7, 8, 9, 10}, c[10];
    int i, k;

    for (i=0; i<5; i++) {
        c[i+k] = a[i];
        c[i+k] = b[i];
    }

    for (i=0; i<9; i++) {
        if (a[i] > c[i+1])
            {
                t = c[i];
                c[i] = c[i+1];
                c[i+1] = t;
            }
    }

    for (i=0; i<9; i++)
    {
        printf("%d", a[i]);
    }
}
```