	R V College of Engineering **Department of Computer Science and Engineering** **CIE - I: Question Paper**		
Course: (Code)	**IOT & Embedded Computing (CS344AI)**	**Semester : 4th semester**	
Date : June 2024	**Duration :** 90 Minutes	**Staff :** KB/MSS / SDV/MH	
Name:	**USN** :	**Section :**	A/B/C/D/CD/CY

PART B

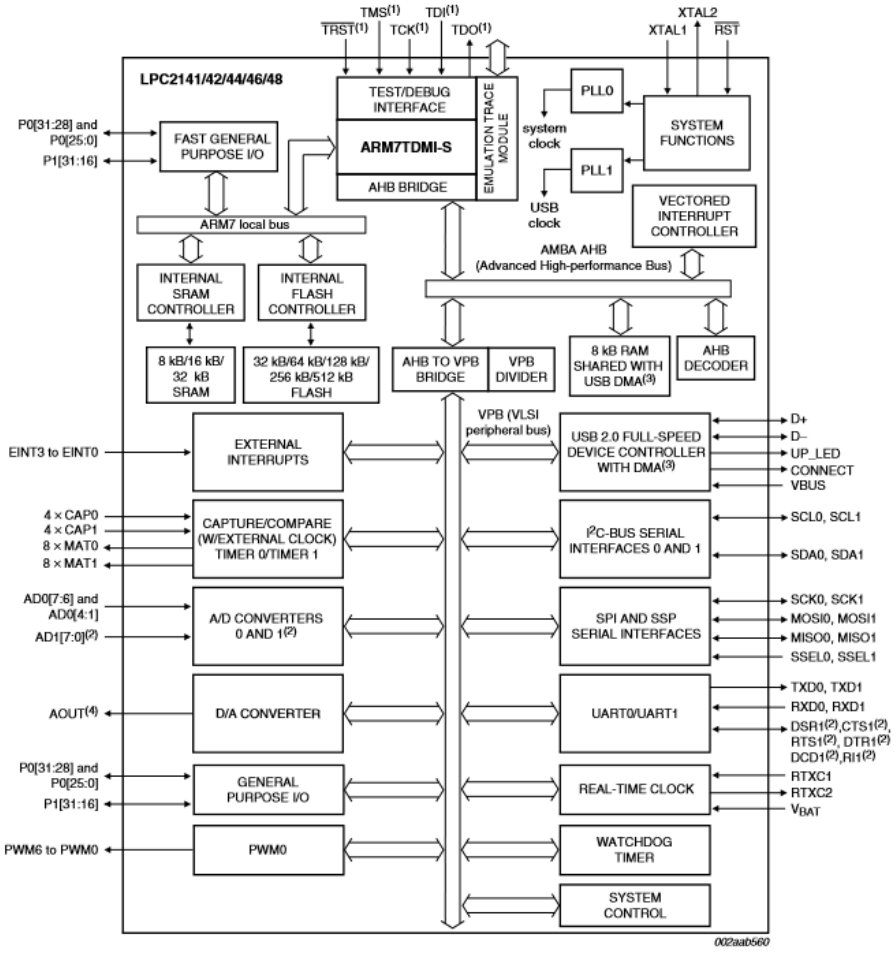
1	With neat Block diagram explain the LPC2148 architecture. List the Peripherals associated and their corresponding applications.	10	L2	CO2
2	a) List the differences between the General-Purpose computing systems and Embedded systems. b) Explain the Operating Modes of ARM using the Register Architecture	10	L3	CO2
3	Interface 5-digit seven segment display to LPC 2148 and write an embedded C program to display the moving string "IOT BOARD".	10	L3	CO3
4	Design a Bank locker system as per the specifications given below by clearly indicating the interface diagram and embedded C code. Requirements: a) Use LPC 2148 Microcontroller and suitable interfacing components. b) Enter a 4digit key to open the locker, If the key entered was correct open the locker door, driven by stepper motor. c) Provide a Key, to close the door. Make suitable assumptions.	10	L4	CO3
5	Explain the working of DAC module of LPC 2148 Microcontroller, and indicate the Resolution, input and output ranges. Write an embedded C program to generate triangular, staircase and rectangular waveforms.	10	L3	CO3

Course Outcomes: After completing the course, the students will be able to:-	
CO 1	Apply Embedded System and IoT fundamentals and formulate sustainable societal relevant cost effective solutions.
CO 2	Demonstrate the development of software programs using Embedded C, using Microcontrollers and different sensors and peripherals to build embedded system applications.
CO3	Design smart systems using various I/O peripherals, Sensors, embedded protocols like UART,I2C,SPI using modern tools like Keil IDE software for various domains like Healthcare, automation, agriculture, smart cities and others.
CO 4	Indulge in developing Novel multi-disciplinary IoT projects using prototype boards, with effective oral & written communication skills and working in teams.
CO 5	Engage in Lifelong Learning by investigating and executing real world societal problems using engineering tools – Cross compilers, debuggers and simulators, emerging processor and controller-based hardware platforms, IOT cloud infrastructure & protocols.

BT LEVELS	L1	L2	L3	L4	L5	L6	COS	CO1	CO2	CO3	CO4
MARKS		10	30	10					20	30	

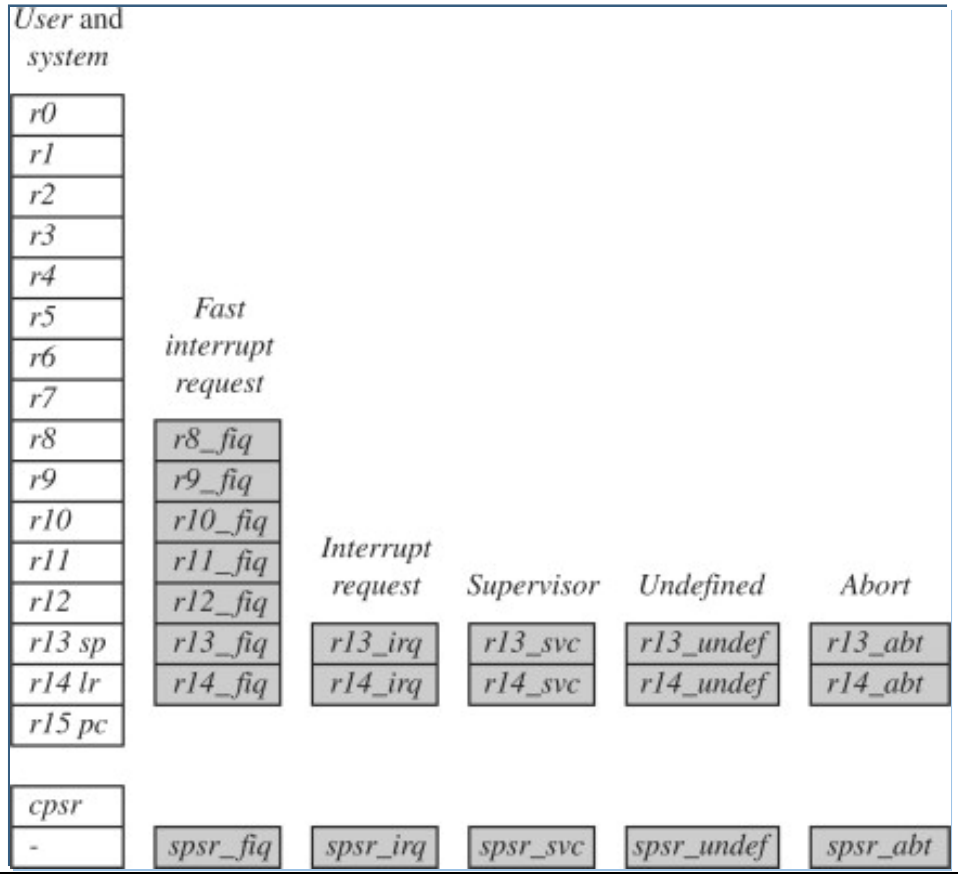
	R V College of Engineering **Department of Computer Science and Engineering** **CIE - I: Scheme**	
Course: (Code)	**IOT & Embedded Computing (CS344AI)**	**Semester : 4th semester**

PART B

1	Block diagram – 4 marks Listing peripherals and corresponding Applications – 6 Marks 	10	L2	CO2
2	Each sub question carries 5 marks 1.	10	L3	CO2

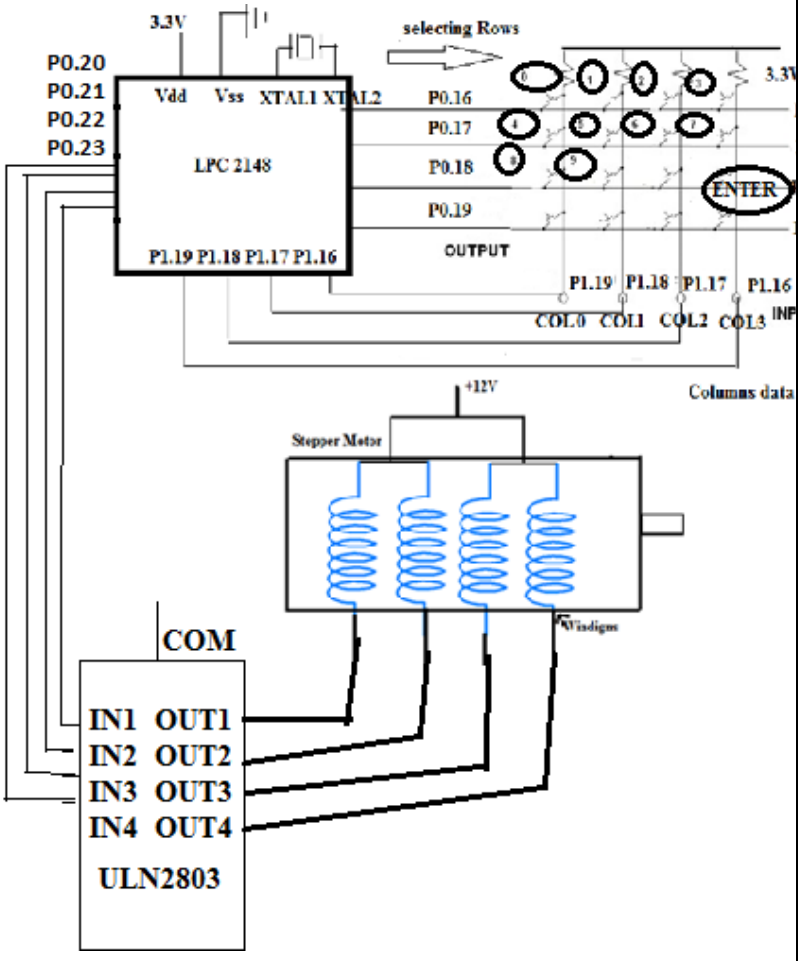
Criteria	General Purpose Computing System	Embedded System
Contents	A system which is a combination of a generic hardware and a General Purpose Operating System for executing a variety of applications.	A system which is a combination of special purpose hardware and embedded OS for executing a specific set of applications.
OS	It contains a general purpose operating system (GPOS).	It may or not contain an operating system for functioning.
Alterations	Applications are alterable (programmable) by the user. (It is possible for the end user to re-install the OS and also add or remove user applications.)	The firmware of the embedded system is pre-programmed and it is non-alterable by the end-user.
Key factor	Performance is the key deciding factor in the selection of the system. Faster is better.	Application specific requirements (like performance, power requirements, memory usage, etc.) are key deciding factors.
Power Consumption	More	Less
Response Time	Not critical	Critical for some applications
Execution	Need not be deterministic	Deterministic for certain types of ES like 'Hard Real Time' systems.

2. Explain the registers corresponding the ARM modes of operating



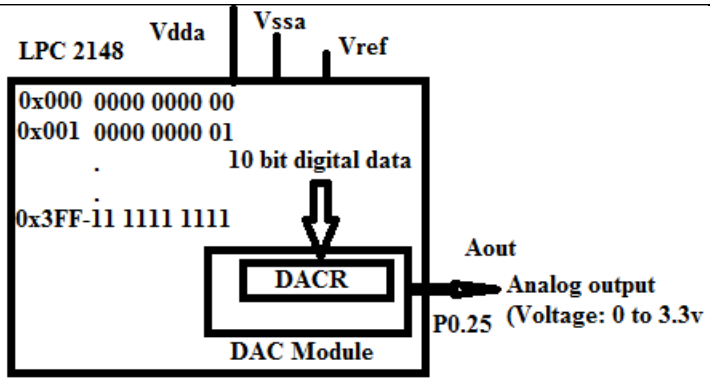
3	*Seven Segment Display Program:* *//P0.19 Data pin of 1st shift register* *//P0.20 Clock pin of shift registers, make 1 to 0* *//P0.30 Strobe pin of shift registers: 1 to 0* #include <lpc214x.h> #define LED_OFF (IO0SET = 1U << 31) #define LED_ON (IO0CLR = 1U << 31) #define PLOCK 0x00000400 void delay_ms(unsigned int j); void SystemInit(void); unsigned char getAlphaCode(unsigned char alphachar);	10	L3	CO3

```
void alphadisp7SEG(char *buf);
int main()
{
IO0DIR |= 1U << 31 | 1U << 19 | 1U << 20 | 1U << 30 ; // to set as o/ps
LED_ON; // make D7 Led on .. just indicate the program is running
while(1)
{
alphadisp7SEG("fire ");
delay_ms(500);
alphadisp7SEG("help ");
delay_ms(500);
}
}
unsigned char getAlphaCode(unsigned char alphachar)
{
switch (alphachar)
{
// dp g f e d c b a - common anode: 0 segment on, 1 segment off
case 'I': return 0xf9;
case 'O': return 0xc0;
case 'T': return 0x93;
case 'B':return ox80;
case 'O':return 0xc0;
case 'A': return 0xf7;
case 'R':return 0xf7;
case 'D':return 0xa1;
case ' ': return 0xff;
//simmilarly add for other digit/characters
default : break;
}
return 0xff;
}
void alphadisp7SEG(char *buf)
{
unsigned char i,j;
unsigned char seg7_data,temp=0;
for(i=0;i<5;i++) // because only 5 seven segment digits are present
{
seg7_data = getAlphaCode(*(buf+i));
// instead of this look up table can be used
// to shift the segment data(8bits)to the hardware (shift registers) using
Data,Clock,Strobe
for (j=0 ; j<8; j++)
{
//get one bit of data for serial sending
temp = seg7_data & 0x80; // shift data from Most significan bit (D7)
if(temp == 0x80)
IOSET0 |= 1 << 19; //IOSET0 | 0x00080000;
else
IOCLR0 |= 1 << 19; //IOCLR0 | 0x00080000;
//send one clock pulse
IOSET0 |= 1 << 20; //IOSET0 | 0x00100000;
```

	delay_ms(1); IOCLR0 |= 1 << 20; *//IOCLR0 | 0x00100000;* seg7_data = seg7_data << 1; *// get next bit into D7 position* } } *// send the strobe signal* IOSET0 |= 1 << 30; //IOSET0 | 0x40000000; delay_ms(1); //nop(); IOCLR0 |= 1 << 30; *//IOCLR0 | 0x40000000;* return; } void delay_ms(unsigned int j) { unsigned int x,i; for(i=0;i<j;i++) { for(x=0; x<10000; x++); } }			
4	 #include <lpc214x.h>	10	L4	CO3

```
#include <string.h>

#define COL0 (IO1PIN & 1<<19)
#define COL1 (IO1PIN & 1<<18)
#define COL2 (IO1PIN & 1<<17)
#define COL3 (IO1PIN & 1<<16)

#define LED_ON (IO0CLR = 1U<<31)
#define LED_OFF (IO0SET = 1U<<31)
#define ENTER   10

void delay_ms(unsigned int);
char getKey(void);
void open(void);  // to open the door
void close(void); // to close the door


char ch,keys[5],password[5] = "0123";
unsigned char len = 0;

unsigned int i = 0;
int main ( )  {
        char ch;
        IO0DIR |=| 0x0f<<16 ;

        do
         {
             i = 0;
            // read the password
            while (1)
             {
                if ((ch = getKey()) == ENTER) break;
                keys[i++]=ch;
             }
            keys[i] ='\0';; // null character, to make it string
            if ( strcmp (keys, password) ==0 )
                {
                  open( ); // rotate clockwise for 90 degree, open the door

                    //Wait for a key 'b' to close the door
                  While ( ( ch = getKey ( ) ) != 'a')  { } ;
                  close( );// rotate anticlockwise for 90 degree, close the
        door

                }

         }while(1);

}


void delay_ms(unsigned int ms){
        unsigned int x, i;
        for(x = 0; x < ms; x++)
                for(i = 0; i < 10000; i++);
```

	} char getKey( ) { unsigned char lookup_table[4][4]={ {'0', '1', '2','3'}, {'4', '5', '6','7'}, {'8', '9', 'a',10}, {'c', 'd', 'e','f'}}; unsigned char rowsel=0,colsel=0; while(1) { //check for keypress in row0,make row0 '0',row1=row2=row3='1' rowsel=0;IO0SET = 0X000F0000;IO0CLR = 1 << 16; if(COL0==0){colsel=0;break;};if(COL1==0){colsel=1;break;}; if(COL2==0){colsel=2;break;};if(COL3==0){colsel=3;break;}; //check for keypress in other rows delay_ms(50); // debouncing delay // wait for a key release while(COL0==0 | COL1==0 | COL2==0 | COL3==0); delay_ms(50); // debouncing delay return lookup_table[rowsel][colsel]; } void open ( ){ for (int i = 0; i < 20; i++) { IO0CLR = 0X000F0000; IO0SET = 0X00080000; delay_ms(15); IO0CLR = 0X000F0000; IO0SET = 0X00040000; delay_ms(15); IO0CLR = 0X000F0000; IO0SET = 0X00020000; delay_ms(15); IO0CLR = 0X000F0000; IO0SET = 0X00010000; delay_ms(15); } IO0CLR = 0x00ff0000; } void close ( ) { for (int i = 0; i < 20; i++) { IO0CLR = 0X000F0000; IO0SET = 0X00010000; delay_ms(15); IO0CLR = 0X000F0000; IO0SET = 0X00020000; delay_ms(15); IO0CLR = 0X000F0000; IO0SET = 0X00040000; delay_ms(15); IO0CLR = 0X000F0000; IO0SET = 0X00080000; delay_ms(15); } IO0CLR = 0x00ff0000; }			
5	DAC Module of LPC 2148: LPC 2148, provides in-built 10-bit Digital to Analog Converter, as shown in the figure below.	10	L3	CO3



DAC module of LPC 2148 is a 10 bit Digital to Analog converter used to convert 10 bit Digital data to corresponding Analog voltage.

 Digital I/P : **000 to 3FF (0 to 1023)**, corresponding Analog O/P : **0V to 3.3V**

 Resolution = (3.3/1024) ≈ **3.2mili volts**

```
#include <lpc214x.h>
#include <stdio.h>
#define SW2 (IO0PIN & (1 << 14))
#define SW3 (IO0PIN & (1 << 15))
#define SW4 (IO1PIN & (1 << 18))
#define SW5 (IO1PIN & (1 << 19))
#define SW6 (IO1PIN & (1 << 20))
static void delay_ms(unsigned int j); //millisecond delay
short int sine_table[ ] =
{512+0,512+53,512+106,512+158,512+208,512+256,512+300,512+342,512+380,512+413,512+442,512+467,512+486,512+503,512+510,512+511,
512+510,512+503,512+486,512+467,512+442,512+413,512+380,512+342,512+300,512+256,512+208,512+158,512+106,512+53,512+0,
512-53,512-106,512-158,512-208,512-256,512-300,512-342,512-380,512-413,512-442,512-467,512-486,512-503,512-510,512-511,
512-510,512-503,512-486,512-467,512-442,512-413,512-380,512-342,512-300,512-256,512-208,512-158,512-106,512-53};
short int sine_rect_table[ ] =
{512+0,512+53,512+106,512+158,512+208,512+256,512+300,512+342,512+380,512+413,512+442,512+467,512+486,512+503,512+510,512+511,
512+510,512+503,512+486,512+467,512+442,512+413,512+380,512+342,512+300,512+256,512+208,512+158,512+106,512+53,512+0};
int main()
{
short int value,i=0;
PINSEL1 |= 0x00080000; /* P0.25 as DAC output :option 3 - 10
While(1){
if ( !SW4) /* If switch for triangular wave is pressed */
{
value = 0;
while ( value != 1023 )
{
DACR = ( (1<<16) | (value<<6) );
```

	value++; } while (value != 0) { DACR = ((1<<16) | (value<<6)); value--; } } void delay_ms(unsigned int j) { unsigned int x,i; for(i=0;i<j;i++) { for(x=0; x<10000; x++); } }			

Course Outcomes: After completing the course, the students will be able to:-	
CO 1	Apply Embedded System and IoT fundamentals and formulate sustainable societal relevant cost effective solutions.
CO 2	Demonstrate the development of software programs using Embedded C, using Microcontrollers and different sensors and peripherals to build embedded system applications.
CO3	Design smart systems using various I/O peripherals, Sensors, embedded protocols like UART,I2C,SPI using modern tools like Keil IDE software for various domains like Healthcare, automation, agriculture, smart cities and others.
CO 4	Indulge in developing Novel multi-disciplinary IoT projects using prototype boards, with effective oral & written communication skills and working in teams.
CO 5	Engage in Lifelong Learning by investigating and executing real world societal problems using engineering tools – Cross compilers, debuggers and simulators, emerging processor and controller-based hardware platforms, IOT cloud infrastructure & protocols.

BT LEVELS	L1	L2	L3	L4	L5	L6	COS	CO1	CO2	CO3	CO4
MARKS		10	30	10					20	30	

	RV College of Engineering® Department of Computer Science and Engineering CIE - II: Test and Quiz Paper		
Course & Code	IOT and Embedded Computing (CS344AI)		Semester: 4th Sem BE
Date : July 2024	Duration:120 minutes	Max.Marks:(10+50)=60 Marks	Staff : KB, SDV, MSS, MH
USN :	Name :		Section : A/B/C/D/CD/CY

NOTE: *Answer all the questions from Part-A (10 M) and Part-B (50 M)*

Sl.no	PART - A	Marks	BT	CO
1	Indicate the value to be loaded into match Register MR0, so that timer counter T0TC reaches the MR0 value after 5 milliseconds. Assume the PCLK = 10MHz, CCLK=40MHz, T0TC=0, Pre-scaler Register=0 (Show the calculations)	2	L3	CO3
2	Calculate the delay produced by the following program run on LPC2148. Given PCLK = 15MHz. Write the answer in milli-seconds. Justify your answer. void delay(void) { T0MCR = 0X04; T0TC = 0X00; T0MR0 = 75000; T0TCR = 0X01; while(T0TC != T0MR0); T0TCR = 0X02; }	2	L3	CO3
3	Given PCLK=15MHz, Required baud rate=9600, Compute the values of DLM:DLL. (Assume DivVal=0, MulVal=1). Show the calculations	2	L3	CO3
4	What are the different types of communication models used in IoT.	2	L2	CO4
5	List any four most commonly used sensors in IoT and mention any two applications of PWM in IoT	2	L2	CO4

Sl.no	PART - B	Marks	BT	*CO
1a.	Generate the 200KHz, 25% duty cycle waveform using LPC 2148 PWM channel. Assume PCLK = 15MHz. Make suitable assumptions, and explain clearly the calculations and the working of the program.	5	L3	CO3
1b.	Generate the 10KHz square waveform using LPC 2148 GPIO pin P0.1. Use timers to calculate the timings and assume PCLK = 60MHz. Explain the working of the program.	5	L3	CO3
2a.	Design an activity LED (one which is blinking once in 10 seconds to indicate the system/product is working) using interrupts and timers, with suitable comments.	5	L3	CO3
2b	Discuss the Features and Applications of serial protocols I2C and SPI.	5	L2	C03
3a	Define IoT and Explain the functional blocks of IoT with the help of neat block diagram.	5	L2	CO3
3b.	Suggest (With brief description) any one-use case of IOT pertaining to following domains: Energy, Retail, Logistics, Agriculture, Cities.	5	L4	CO4
4a.	Design an IOT Level 2 deployment application for weather monitoring and Device control in the house using ESP32 and Thing speak cloud platform, with suitable block diagram, interfacing, flowcharts and brief description. The proposed system consists of single node that monitors the room temperature and humidity using DHT 11 sensor, and based on the temperature / humidity, device(fan) should be turned on using a Relay. The controller also sends the sensor data to the cloud, where it will be displayed on the dash board.	5	L6	CO4
4b.	Design an IOT Leve2 deployment application for Smart Parking using RasberryPie with IR sensors and Cloud with Mobile Application to show the parking slots status. Draw the block diagram, interfacing, flowchart and brief description.	5	L6	CO4
5	Interface LDR and LED bulb to LPC 2148 and write an embedded C program to read the data from LDR and suitably turn on/off the LED bulb and also send the suitable message to computer using UART interface. Clearly show the connections between LPC 2148 and Computer Serial Port and explain the UART initialization steps, clearly showing the registers used and the baud rate calculations.	10	L3	CO3

BT LEVELS	L1	L2	L3	L4	L5	L6	COS	CO1	CO2	CO3	CO4
MARKS		14	31	5		10				41	19

	RV College of Engineering® Department of Computer Science and Engineering CIE - II: Test and Quiz Paper		
Course & Code	IOT and Embedded Computing (CS344AI)		Semester: 4th Sem BE
Date : July 2024	Duration:120 minutes	Max.Marks:(10+50)=60 Marks	Staff : KB, SDV, MSS, MH
USN :	Name :		Section : A/B/C/D/CD/CY

NOTE: *Answer all the questions from Part-A (10 M) and Part-B (50 M)*

Sl.no	PART - A	Marks	* BT	*CO
1	Indicate the value to be loaded into match Register MR0, so that timer counter T0TC reaches the MR0 value after 5 milliseconds. Assume the PCLK = 10MHz, CCLK=40MHz, T0TC=0, Pre-scaler Register=0 Ans: 50000	2	L2	CO3
2	Calculate the delay produced by the following program run on LPC2148. Given PCLK = 15MHz. Choose the answer in milli-seconds. void delay(void) { T0MCR = 0X04; T0TC = 0X00; T0MR0 = 75000; T0TCR = 0X01; while(T0TC != T0MR0); T0TCR = 0X02; } Ans: 5ms	2	L3	CO2
3	Given PCLK=15MHz, Required baud rate=9600, Choose the values of DLM:DLL. (Assume DivVal=0, MulVal=1) Ans: U0DLM=00;U0DLL=97;	2	L2	CO2
4	What are the different types of communication models used in IoT.	2	L2	CO2

	Ans:· Request-Response Communication Model · Publisher-Subscriber Communication Model · Push-Pull Communication Model · Exclusive-Pair Model			
5	List any four most commonly used sensors in IoT and mention any two applications of PWM in IoT Ans: Sensors- Temperature, Humidity, Moisture, Air Pollution, Vibration PWM Applications: LED Lighting, Servo Motor Control, DC Motor Control	2	L3	CO3

Sl.no.	PART - B	Marks	* BT	*CO
1a.	Generate the 200KHz, 25% duty cycle waveform using LPC 2148 PWM channel. Assume PCLK = 15MHz. Make suitable assumptions, and explain clearly the calculations and the working of the program. Assume PCLK = 15MHz T1 = Time Period of 200KHz = 1/20KHz = 0.005 msec T2 = Time Period of PCLK = 1/PCLK = 0.067 Microsecs No. of PCLKs required for one Timer period of 0.05ms= T1/T2 = 0.05msec/0.067Microsec= 74 to be loaded in MR0 register Assume PWM3 and PWM6 are used for generating waveforms with different duty cycle ratios, MR3 = 0.25 x 74 = 18 (25 % duty cycle) #include <LPC214x.h> void PWM_Init(void) #include <LPC214x.h> void PWM_Init(void)	5	L2	CO2

	{ //P0.1 pin has second alternate function as PWM3 channel, so using PINSEL0 register PINSEL0 |= 0x00000008; // Select P0.1 as PWM output , bits D2 & D3 are for P0.1 PINSEL0 |= 2 << 18; *//select P0.9 as PWM6 (option 2)* //Configure PWM channel 3 & 6 as single edge type and enable the channel PWMPCR = (1<<11) | (1 << 14); //load the value to MR0 to fix the pulse rate PWMMR0 = 74; // 200KHz pulse rate // enable PWM unit of LPC2148 and start the timer PWMTCR = 0x0000 0009; // bit D3 = 1 (enable PWM), bit D0=1 (start the timer) } int main() { PWM_Init(); while(1) { PWMMR3 = 18; //25% duty cycle PWMLER = 0X48; // enable for channel 3 and 6 } }			
1b.	Generate the 10KHz square waveform using LPC 2148 GPIO pin P0.1. Use timers to calculate the timings and assume PCLK = 60MHz. Explain the working of the program	5	L2	CO2

Td = 1/10KHz = 0.1 msec , half of it is 0.05msec;

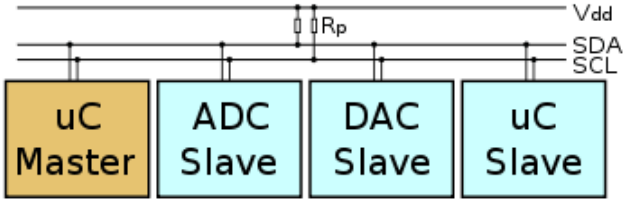
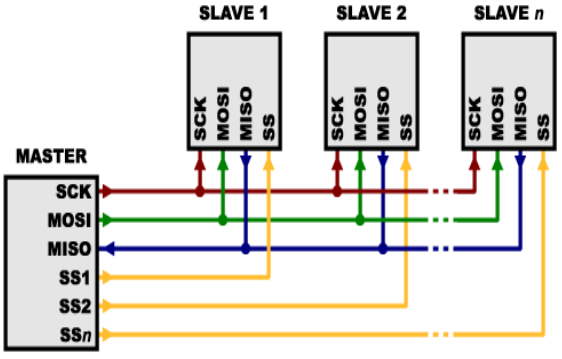
T = 1 / PCLK = 1 / (60MHz) = 0.01666

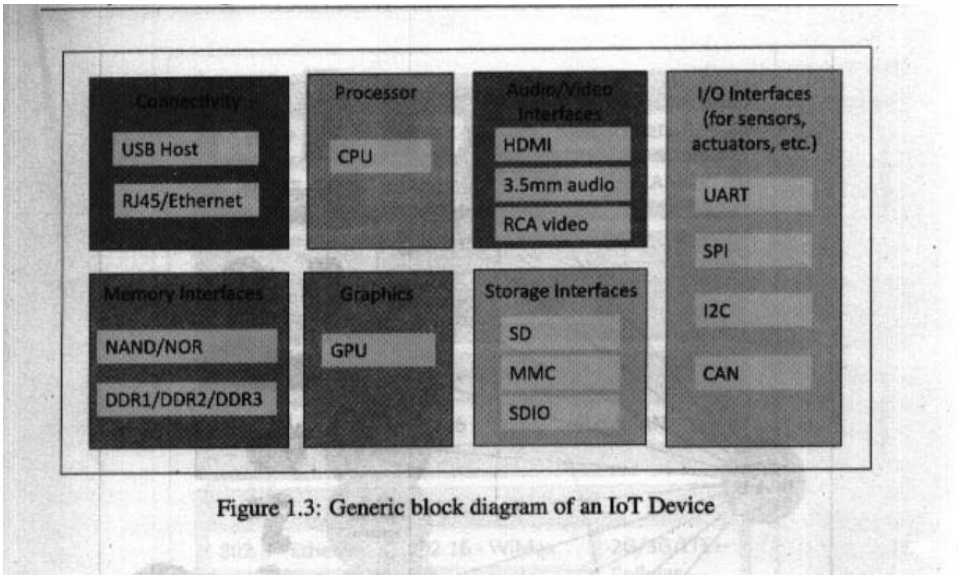
micro seconds
count =

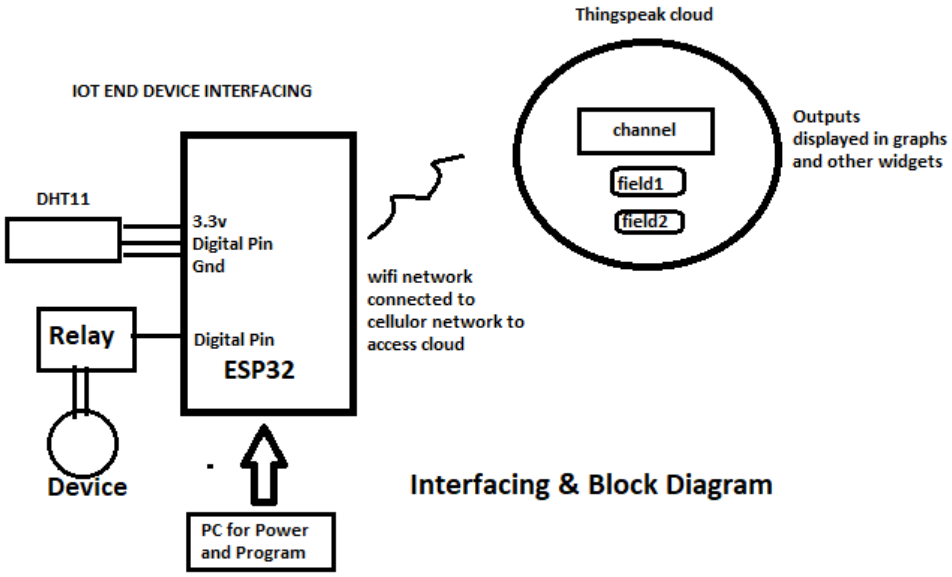
Td / T = 0.05 msec/0.0166 micro = 3000

```
int main(void)
{
    T0MR0 = 3000 ;  //use the Timer0 and
load the MR0 with count
T0MCR = 0X0004;  //   0000….100 – Stop
the timer, after match
    I0DIR0 = 0X00000002; //make P0.1 as output
while(1)  // program to produce square
waveform  of 1 KHz
            {
            I0SET0
  =  1 << 1;  //set P0.1 to 1
            delay(
);
            I0CLR0
=  1 <<1;  //clear P0.1 to 0
delay(
);
            }
```

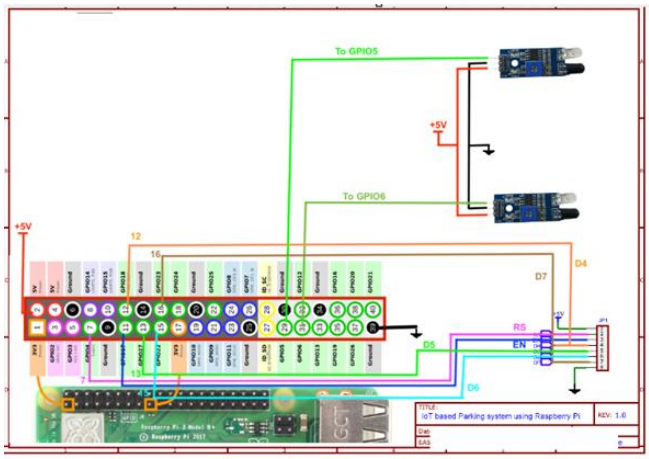
	} void delay(void) { T0TCR = 1; //start the timer While (!(T0TC == T0MR0)); T0TCR = 2; // reset the counter and stop the timer }			
2a.	**Design an activity LED (one which is blinking once in 10 seconds to indicate the system/product is working) using interrupts and timers, with suitable comments** #include <LPC2148x.h> unsigned int x=0; __irq void Timer0_ISR(void) // an ISR program { x = x ^ 1; if (x) I0SET1 = 1 << 16; //P1.16 = 1 else I0CLR1 = 1 <<16; // P1.16 = 0 T0IR = 0x01; // clear match0 interrupt, and get ready for the next interrupt VICVectAddr = 0x00000000 ; //End of interrupt } int main(void) { I0DIR1 = 0x0001 0000; //set P1.16 as output T0TCR = 0x00; // stop the timer, to initialize different registers T0MCR= 0x0003; // Enable Interrupt and reset timer after match T0TC = 0x00; // make TC = 0 T0MR0 = 150000; // generates 10ms //load interrupt related registers , assigning Timer0 to IRQ slot 4 VICVectAdd4 = (unsigned long)Timer0_ISR; // set the timer ISR vector address VICVectCntl4 = 0x0000024; // set the channel VICIntEnable = 0x0000010; // enable the timer0 interrupt T0TCR = 0x01; // start the timer while(1) { //do other works	5	L2	CO3

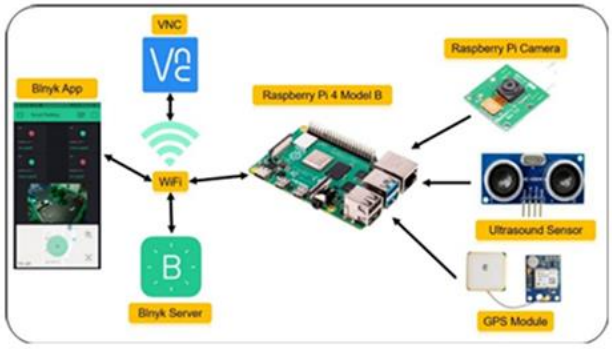
	}; // now timer interrupt is serviced automatically using the ISR }			
2b.	Discuss the Features and Applications of serial protocols I2C and SPI  **I2C Features** ▪ It is multi-master, multi-slave, packet switched, single-ended, serial computer bus. It is widely used for attaching lower-speed peripheral IC's to processors and microcontrollers in short-distance, intra board communication. ▪ Support multi master system, more than one master can communicate with the devices, for every 8 bits of data sent, one extra bit of meta data (ACK/NACK bit) must be transmitted. ACK/NACK bit gives confirmation that each frame is transferred successfully. ▪ Only uses two wires (Serial, half duplex), Hardware is less complicated than with UARTs. The hardware required to implement I2C is more complex than SPI but simpler than UART. ▪ It Supports 128 devices (7bit address) in normal mode. ▪ Data transfer rates up to 100 kbits/s and 7-bit addressing possible in normal mode. (It supports 400Khz (fast mode), 1Mhz –fast mode plus,3.4Mhz for high speed mode, 5Mhz for ultra fast mode) **SPI (Note: connections for one device good enough)**  ➢ SPI Interface uses four wires for communication. Hence it is also known as four wire serial communication protocol.	5	L3	CO1

	➢ SPI is a full duplex master-slave communication protocol. This means that only a single master and a single slave can communicate on the interface bus at the same time. It has separate send & receive lines unlike I2C. ➢ SPI enabled devices work in two basic modes of SPI operation i.e. SPI Master Mode and SPI Slave Mode. Master Device is responsible for initiation of communication. Master Device generates Serial Clock for synchronous data transfer. There is always only one master (most of the times it is microcontroller). ➢ Faster than asynchronous serial (UART), operate around 1Mhz. (can go upto 10Mhz) ➢ Hardware requirement for SPI is very simple (as simple as shift register) compare to UART & I2C. ➢ Master Device can handle multiple slave devices on the bus by selecting them one by one using multiple slave select pins. In general, each slave will need a separate SS line.			
3a.	Define IoT and Explain the functional blocks of IoT with the help of a neat block diagram. Definition of IoT - 1 Mark, Block diagram - 2 Marks, Brief explanation-2 marks  Figure 1.3: Generic block diagram of an IoT Device	5	L3	CO3
3b.	Suggest (With brief description) any one-use case of IOT pertaining to following domains: Energy, Retail, Logistics, Agriculture, Cities. Any one-use case: Block diagram representation (optional) + Explanation	5	L4	CO1

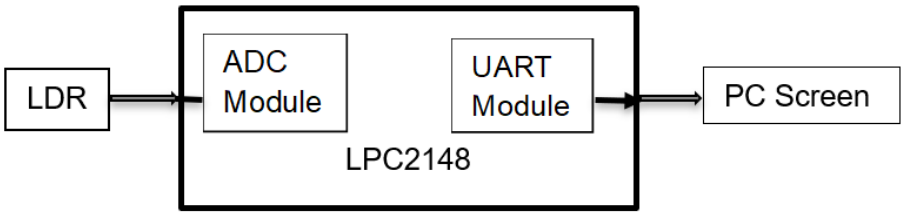
4a.	Design an IOT Level 2 deployment application for weather monitoring and Device control in the house using ESP32 and Thing speak cloud platform, with suitable block diagram, interfacing, flowcharts and brief description. The proposed system consists of single node that monitors the room temperature and humidity using DHT 11 sensor, and based on the temperature / humidity, device(fan) should be turned on using a Relay. The controller also sends the sensor data to the cloud, where it will be displayed on the dash board.  Interfacing & Block Diagram **Algorithm:** On the Cloud: 1.Goto thingspeak cloud, do following things: Create the channel and two fields for storing temperature and humidity 2.Copy the Channel no and write API to be used at the IOT Edge device. On the End Device: 1. Import the Libraries for DHT11,ThingSpeak cloud 2. Initialize the required variables, libraries (start functions) with appropriate digital pins used for interfacing. Set the temperature threshold for fan/device control 3. Read the temperature and humidity 4. Upload the values to the thingspeak cloud channel fields, using the channel ID and fields. 5. Compare the room temperature with threshold and switch on / off the relay to control the device/fan. 6. delay, after each reading	10	L4	CO2

	7. repeat the steps 3 - 6			
4b.	Design an IOT Leve2 deployment application for Smart Parking using RasberryPie with IR sensors and Cloud with Mobile Application to show the parking slots status. Draw the block diagram, interfacing, flowchart and brief description. **Firebase Configuration:** Configuring Firebase involved several steps: ● In the Project settings under the Firebase Admin SDK section, we generated a new private key and saved the key.json file to our project directory. This key was essential for authenticating and interacting with Firebase services. Firebase was used to store real-time data on parking spot availability, manage user authentication, and log vehicle entry and exit times. Component Setup: ● Raspberry Pi 4: Served as the central hub for managing the parking system, running the application, and interfacing with hardware components. ● Ultrasonic Sensors: Installed at each parking spot to detect the presence of vehicles. These sensors were connected to the Raspberry Pi GPIO pins. ● Camera Module: Used to capture images of vehicles entering and exiting the parking area for license plate recognition. ● LED Indicators: Installed to show the status of each parking spot (occupied or available). ● Display Screen: Provided real-time information on parking availability to users at the entry point. Hardware Configuration: Raspberry Pi GPIOs are connected to the ultrasonic sensors and LED indicators. The pinout diagram was essential to ensure the connections were correct and avoid any potential hardware damage. The camera module is secured and positioned to capture clear images of vehicle license plates.	5	L4	CO2

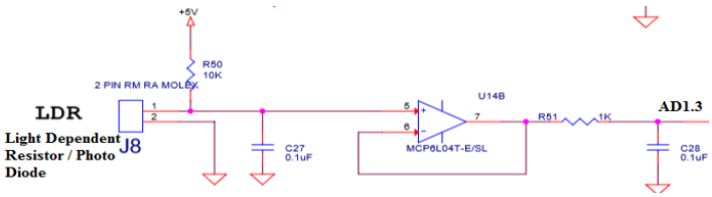




5 Interface LDR and LED bulb to LPC 2148 and write an embedded C program to read the data from LDR and suitably turn on/off the LED bulb and also send the suitable message to computer using UART interface. Clearly show the connections between LPC 2148 and Computer Serial Port and explain the UART initialization steps, clearly showing the registers used and the baud rate calculations.



LDR:



AD0CR=0x00200600|(1<<ch); *//select channel*

```
    AD0CR|=(1<<24);                    //start conversion
    while((AD0GDR& (1U<<31))==0);
    val=AD0GDR;

  U0THR =val;  //send to serial port(check on the terminal)

UART Initialization::
void uart_init(void)
{
  //configurations to use serial port
 PINSEL0 |= 0x00000005;  // P0.0 & P0.1 ARE CONFIGURED AS TXD0
& RXD0
 U0LCR = 0x83;   /* 8 bits, no Parity, 1 Stop bit       */
 U0DLM = 0; U0DLL = 8; // 115200 baud rate
 U0LCR = 0x03;  /* DLAB = 0                  */
 U0FCR = 0x07;  /* Enable and reset TX and RX FIFO. */
}
```

Course Outcomes: After completing the course, the students will be able to:-	
CO 1	Apply Embedded System and IoT fundamentals and formulate sustainable societal relevant cost effective solutions.
CO 2	Demonstrate the development of software programs using Embedded C, using Microcontrollers and different sensors and peripherals to build embedded system applications.
CO3	Design smart systems using various I/O peripherals, Sensors, embedded protocols like UART,I2C,SPI using modern tools like Keil IDE software for various domains like Healthcare, automation, agriculture, smart cities and others.
CO 4	Indulge in developing Novel multi-disciplinary IoT projects using prototype boards, with effective oral & written communication skills and working in teams.
CO 5	Engage in Lifelong Learning by investigating and executing real world societal problems using engineering tools – Cross compilers, debuggers and simulators, emerging processor and controller-based hardware platforms, IOT cloud infrastructure & protocols.

BT LEVELS	L1	L2	L3	L4	L5	L6	COS	CO1	CO2	CO3	CO4
MARKS		10	30	10					20	30	

	RV College of Engineering® Department of Computer Science and Engineering Improvement Test and Quiz Paper		
Course & Code	IOT and Embedded Computing (CS344AI)		Semester: 4th Sem BE
Date : Aug 2024	Duration:120 minutes	Max.Marks:(10+50)=60 Marks	Staff : KB, SDV, MSS, MH
USN :	Name :		Section : A/B/C/D/CD/CY

NOTE: *Answer all the questions from Part-A (10 M) and Part-B (50 M)*

Sl.no	PART - A	Marks	BT	CO
1	Suggest any one application of Level 5 and Level 6 IOT deployment.	2	L3	CO5
2	Describe an Example of IoT service that uses publish-subscribe communication model. Name the popular application layer protocol for publish-subscribe model used in resource constraint IOT systems.	2	L3	CO5
3	Name the pins provided by RasberryPie to support I2C and SPI interfaces.	2	L2	CO4
4	Evaluate the following statements and indicate whether they are true/false. a) Von Neumann Architecture shares common memory for Data and Instructions b) Harvard Architecture has separate physical memories for Data and Instructions	2	L3	CO1
5	Consider a four-bit ALU which does four bits arithmetic. When the following four-bit numbers are added, what is the status of NZCV flags? 1101 + 1011	2	L4	CO2

Sl.no	PART - B	Marks	BT	*CO
1	Draw the deployment design of the weather monitoring IOT system. Further, show the mapping of IOT Level to Functional Groups for the weather monitoring IoT system.	5	L3	CO5
2	Write the programs to perform the following: (draw interface diagrams) - Interface one LED to GPIO 18, and program for blinking the LED (use RasberryPie and phython)	5	L3	CO4

	- Interface one LDR to D36 and LED to D2, and make the LED on/off based on Light Intensity (use ESP32 and embedded C)			
3	The purpose of the home intrusion detection system is to detect intrusion using sensors (PIR sensor and Door sensor). Design Home Intrusion Detection system using RPie/ESP32 with PIR motion sensor for motion detection and door sensor for detecting opening / closing of the door (for one room). Draw the following (no explanation required) - Process Specification - Domain model - Deployment design - Functional & Operational View specifications	10	L4	CO5
4	a) With a neat diagram explain the architecture of ARM Microcontroller. b) With the neat diagram briefly describe operating modes and register organization of ARM ISA. Mention the use of following Registers: R13,R14,R15,CPSR and SPSR.	5 5	L2 L3	CO1 CO2
5	a) Explain how embedded system are classified. b) Differentiate between RISC and CISC architecture.	5 5	L3 L2	CO2 CO3

CS344AI

USN 1 RV 22 CD 059

RV COLLEGE OF ENGINEERING®

(An Autonomous Institution Affiliated to VTU)

IV Semester B. E. Regular Examinations Sept/Oct – 2024

Common to CS/CY/CD

IOT AND EMBEDDED COMPUTING

Time: 03 Hours *Maximum Marks: 100*

Instructions to candidates:

1. Answer all questions from Part A. Part A questions should be answered in first three pages of the answer book only.
2. Answer FIVE full questions from Part B. In Part B question number 2 is compulsory. Answer any one full question from 3 and 4, 5 and 6, 7 and 8, 9 and 10.

PART-A

			M	BT	CO
1	1.1	Define Pipeline. How stages of pipeline do $ARM7$ Support?	02	1	1
	1.2	Which protocol is used to interface the SD card to the Microcontroller $LPC2148$?	02	1	2
	1.3	Given $PCLK = 15\,MHz$, required baud rate= 9600. Compute the values of $DLM:DLL$. (Assume $DivVal = 0, MulVal = 1$). Show the calculations.	02	2	2
	1.4	Write an Embedded C code to make common anode LED connected to $P0.31$ ON and cathode LED connected to $P0.28$ OFF	02	2	3
	1.5	Indicate the value to be loaded into match Register $MR0$, so that time counter $T0TC$ reaches the $MR0$ value after 10 milliseconds. Assume the $PCLK = 10\,MHz, CCLK = 40MHz, T0TC = 0$, Prescaler Register = 0.	02	3	3
	1.6	Write the five pins used on Raspberry pi for SPI interface.	02	1	4
	1.7	Describe the purpose and behavior of Smart home automation by using the standard IoT design methodology.	02	2	4
	1.8	What is the role of Things and Internet in IoT?	02	1	4
	1.9	List any four applications of IoT for logistics.	02	1	3
	1.10	List any four most commonly used sensors in IoT and mention any two applications of PWM in IoT.	02	2	2

PART-B

2	a	With the help of a neat block diagram of $LPC2148$, indicate the different peripheral blocks present inside the controller and their application.	10	2	2
	b	List the differences between the General–Purpose computing systems and Embedded systems.	06	2	1
3	a	Write an embedded C program to interface 4×4 matrix keyboard using Lookup table and display the key pressed on the terminal.	10	3	3
	b	Write a C program to display message "$RVCE$" and "CSE" on 5-digit seven segment display alternatively with a suitable delay.	06	3	3
		OR			
4	a	Explain the working of DAC module of LPC 2148 Microcontroller, and indicate the Resolution, input and output ranges. Write an embedded C program to generate Sine waveform.	10	3	3
	b	Interface $3LEDs$ (Red, Yellow, Green) to $LPC2148$ and write Embedded C program to simulate the traffic light system.	06	3	3

5	a	Explain the concept of *PWM* and its registers. Write an embedded C program to generate *PWM* wave to controls speed of *DC* motor. Control the duty cycle by analog input.	10	3	2
	b	Briefly explain how to interface high power devices using Relays?	06	2	2
		OR			
6	a	Describe the working of *UART* module of *LPC* 2148. Draw the connections between Microcontroller *UART and PC* serial port. Show the baud rate calculations also.	08	3	2
	b	With a neat diagram describe the working of *LCP* 2148 Timers.	08	2	3
7	a	List and explain any two IoT communication models with neat diagrams.	08	2	3
	b	Suggest (with brief description) any one-use case of IoT pertaining to following domains: Retail, Logistics, Agriculture, smart cities.	08	2	3
		OR			
8	a	With suitable block diagrams, explain IoT level 6 and its deployment. Indicate the significance of level 6 deployment.	08	3	4
	b	What is IoT? Explain different characteristics of IoT and their use cases in Industry.	08	2	4
9	a	Consider the Smart Lighting case study and write the following steps of IoT design methodology: i) Purpose and requirements specification ii) Domain model specification iii) Information model specification iv) Service specification	10	4	4
	b	Discuss the features and applications of serial protocols *I2C* and *SPI*.	06	2	4
		OR			
10	a	Design an IoT level 2 deployment application for Smart Parking using Rasberrypie with *IR* sensors and Could with mobile application to show the parking slots status.	06	4	4
	b	The purpose of the Home Intrusion Detection System is to detect Intrusion using sensors (*PIR* sensor and Door sensor). Design Home Intrusion Detection system using *RPie/ESP*32 with *PIR* motion sensor for motion detection and door sensor for detecting opening / closing of the door. Answer the following with necessary design / functional diagram. i) Process specification ii) Domain model iii) Deployment design iv) Functional and operational view specifications.	10	4	4