

CODE OF FIRST ATMEGA

.ORG 0X00

.include "M32DEF.INC"

.org 0x00

.equ LCD\_PRT = PORTA

.equ LCD\_DDR = DDRA

.equ LCD\_PIN = PINA

.equ LCD\_RS = 0

.equ LCD\_RW = 1

.equ LCD\_EN = 2

.EQU KEY\_PORT = PORTC

.EQU KEY\_PIN = PINC

.EQU KEY\_DDR = DDRC

JMP MAIN

.ORG 0X02

JMP KPD\_ISR

.ORG URXCaddr

JMP RXISR

MAIN:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Stack intializaion\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R20, HIGH (RAMEND)

OUT SPH,R20

LDI R20, LOW (RAMEND)

OUT SPL,R20

;intializinting LCD

CALL SERIALCOM\_init

CALL LCD\_Init

LDI R17,0X0F

OUT DDRB,R17 /\* PORT FOR THE STEPPER MOTOR\*/

LDI R19,0X12 /\*\*\*\*\*\*\*\*\*\*\*\*\*R19 IS CHECK FOR CLEARING LCD\*\*\*\*\*\*\*\*/

LDI R28,0X00 /\*\*\*\*\*\*\*\*\*\*\*\*\*\*R28 INTIAL CHECK REGISTER\*\*\*\*\*\*\*\*\*\*\*/

LDI R22,0X00 /\*\*\*\*\*\*\*\*\*\*\*\*R22 IS THE PASSWORD INTAKE FIRST COUNTER\*\*\*\*\*\*\*\*\*\*/

LDI R18,0X01 /\*\*\*\*\*\*\*\*\*\*\* R18 IS THE PASSWORD INTAKE SECOND COUNTER\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R29,0X00 /\*\*\*\*\*\*\*\*\*\*\*\*\*IT CANT BE USED BECAUSE ITS FOR X INDIRECT MODE\*\*\*\*\*\*/

LDI R28,0X00 /\*\*\*\*\*\*\*\*\*\*\*IT CANT BE USED BECAUSE IT IS USING FOR INDIREST MODE Y\*\*\*\*\*\*\*\*\*\*/

LDI R24,0X00 /\*\*\*\*\*\*\*\*\*\* REG 1 USE FOR PASSWORD INTAKE\*\*\*\*\*\*\*/

LDI R25,0X00 /\*\*\*\*\*\*\*\*\*\*REG 2 USE FOR PASSWORD INTAKE\*\*\*\*\*\*\*/

/\*ASSIGNING THE PASSWORD\*/

/\*\*\*\*\*\*\*\*\*\*\*\* ASSIGNING COMPELETE\*\*\*\*\*\*\*\*\*\*\*\*/

; ascii code for keypressed displayed on PORTD

LDI R21, 0xFF

OUT DDRA,R21

; PC0 – PC3 rows

; PC4 – PC7 columns

LDI R20, 0xF0

OUT KEY\_DDR, R20

GroundAllColumns:

LDI R20, 0x0F

OUT KEY\_PORT, R20

/\* THIS IS THE PART PF INIATLIZATION OF THE INTRUPT\*/

LDI R20,0X20

OUT MCUCR,R20

SBI PORTD,2

LDI R20,1<<INT0

OUT GICR,R20

SEI

/\*\*\*\*\*\*\*\*\*fINISHING THE INTIALIZATION OF INTRUPT\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NOW DOING SERIAL COMMUNICATION WITH THE OTHER ATMEGA32\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

AGAIN1:

RJMP AGAIN1

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCTIN DEfinations\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

KPD\_ISR:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CHECKING FOR ERASING lcd \*\*\*\*\*\*\*\*\*\*\*\*\*/

CALL ERASELCD

CALL KEY

call delay

call delay

call delay

call delay

LPM R20, Z /\*\*\*\*\*\*\*\*\*\*\* R20 CONTAIN THE KEY PRESSED\*\*\*\*\*\*\*\*\*/

MOV R16,R20

CALL DATAWRT

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* NOW FIRST CHECKING FOR THE PASSWORD INTAKE FLAG\*\*\*\*\*\*\*\*\*/

CPI R22,0X01

BRNE NOPASSINPUT

;STORING THE VALUE

ST Y, R20

INC YL

;INREAMENT THE COUNTER

INC R23

;CHECKING FOR THE COUNTER

CPI R23,0X3

BRNE RETURN

; CALLING THE FUNCTION TO CHECK THE PASS

CALL PASSCHECK

RETURN:

CALL GROUNDCOL

RETI

NOPASSINPUT:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CHECKING FOR #\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CPI R16,0X23

BRNE INTIALJMP /\* WHEN NOT EQUAL TO JMP TO INTIALJMP\*/

CALL display\_intro

INTIALJMP:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* CHECKING FOR THE CHECKS\*\*\*\*\*\*\*\*\*\*\*\*/

CPI R16,'1' /\*\*\*\*\*\*\*\*\*\* CHECKING TO INPUT PASSWORD\*\*\*\*/

BRNE NOT1

LDI R16,0x01

CALL CMNDWRT

LDI ZH,HIGH(MYDATA2<<1)

LDI ZL,LOW(MYDATA2<<1)

CALL DISPLAYDATA

LDI R16,0XC0

CALL CMNDWRT

LDI R22,0X01/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* THIS MEAN THAT THE INTAKE PASSWORD CHECK IN ON\*\*\*\*\*/

LDI R23,0X00/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*COUNTER FOR THE PASSWORD TAKE\*\*\*\*\*\*\*\*\*\*\*\*/

CALL GROUNDCOL

RETI

NOT1:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* checking for the temp readings\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

CPI R16,'\*'

BRNE NOTEMP

LDI R16,0x01

CALL CMNDWRT

LDI ZH,HIGH(MYDATA4<<1)

LDI ZL,LOW(MYDATA4<<1)

CALL DISPLAYDATA

LDI R16,0XC0

CALL CMNDWRT

CALL SERIALCOM\_init1

LDI R16,'T'

CALL TRANSMIT

CALL SERIALCOM\_init

CALL GROUNDCOL

RETI

NOTEMP:

CALL GROUNDCOL

RETI

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PASSWORD CHANGE\*\*\*\*\*\*\*

CPI R16,'2'\*/

/\* PASSWORD CHECK FUNCTION \*/

PASSCHECK:

DEC YL

SBI PORTA,5

LD R24,Y

LDI ZH,HIGH(DEFPASS<<1)

LDI ZL,LOW(DEFPASS<<1)

LPM R25,Z

CP R24,R25

BRNE NOTEQUAL

CALL INE

CP R24,R25

BRNE NOTEQUAL

CALL INE

CP R24,R25

BRNE NOTEQUAL

LDI R16,0X01

CALL CMNDWRT

LDI ZH,HIGH(WEL<<1)

LDI ZL,LOW(WEL<<1)

CALL DISPLAYDATA

LDI R16,0XC0

CALL CMNDWRT

LDI ZH,HIGH(OS<<1)

LDI ZL,LOW(OS<<1)

CALL DISPLAYDATA

LDI R22,0X00

LDI R18,0X01

CALL RUNMOTOR

RET

NOTEQUAL:

CPI R18,0X03

BRNE NOTALARM

LDI R16,0X01

CALL CMNDWRT

LDI ZH,HIGH(WAR1<<1)

LDI ZL,LOW(WAR1<<1)

CALL DISPLAYDATA

LDI R16,0XC0

CALL CMNDWRT

LDI ZH,HIGH(WAR2<<1)

LDI ZL,LOW(WAR2<<1)

CALL DISPLAYDATA

CALL ALARM

LDI R18,0X01

AGAIN3:

JMP AGAIN3

NOTALARM:

INC R18

LDI R16,0X01

CALL CMNDWRT

LDI ZH,HIGH(AGAIN<<1)

LDI ZL,LOW(AGAIN<<1)

CALL DISPLAYDATA

LDI R16,0XC0

CALL CMNDWRT

LDI R23,0X00

RET

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*///

ALARM:

SBI PORTB,6

CALL SERIALCOM\_init

RET

INE:

DEC YL

INC ZL

LD R24,Y

LPM R25,Z

OUT PORTA,R25

RET

display\_intro:

LDI R28,0X01/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*USING R28 AS A # ENETERED CHECK\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R19,0X01/\*\*\*\*\*\*\*\*\* clearing the display after the screen function\*\*\*\*/

LDI R16,0X01

CALL CMNDWRT

LDI ZH,HIGH(MYDATA<<1)

LDI ZL,LOW(MYDATA<<1)

CALL DISPLAYDATA

LDI R16,0XC0

CALL CMNDWRT

LDI ZH,HIGH(MYDATA1<<1)

LDI ZL,LOW(MYDATA1<<1)

CALL DISPLAYDATA

RET

.ORG 0x500

MYDATA:.DB "ENT1 FR PASS",0

MYDATA1:.DB "ENT2 FR CHG PASS",0

MYDATA2:.DB "ENTER PASS",0

MYDATA3:.DB "ENT CHG PASS",0

MYDATA4:.DB "Temperature",0

CHECK:.DB '0','1'

.ORG 0x300

KCODE0: .DB '1', '4', '7','\*'

KCODE1: .DB '2', '5', '8','0'

KCODE2: .DB '3', '6', '9','#'

DEFPASS:.DB '3','2','1'

WEL:.DB "WELCOME!!!",0

OS:.DB "ENG OSAMA",0

AGAIN:.DB "SORRY ENT AGN",0

WAR1:.DB "!-!\*WHo ARE yOU!!"

WAR2:.DB"!BCHAO CHOR"

WAIT15MS:

PUSH R20

PUSH R21

LDI R21, 0xFF

LOOP3: LDI R20, 0xFF

LOOP2: DEC R20

BRNE LOOP2

DEC R21

BRNE LOOP3

POP R21

POP R20

RET

DISPLAYDATA:

L1:

LPM R20,Z

CPI R20,0

BREQ HERE

MOV R16,R20

CALL DATAWRT

INC ZL

RJMP L1

HERE:

RET

KEY:

LDI R21, 0b10111111

OUT KEY\_PORT,R21

NOP

IN R21,KEY\_PIN

ANDI R21,0x0F /\* EXTRACTING THE ROWS\*/

CPI R21,0x0F

BRNE COL1

LDI R21, 0b11011111

OUT KEY\_PORT, R21

NOP

IN R21, KEY\_PIN

ANDI R21,0x0F

CPI R21,0x0F

BRNE COL2

LDI R21, 0b11101111

OUT KEY\_PORT, R21

NOP

IN R21, KEY\_PIN

ANDI R21,0x0F

CPI R21,0x0F

BRNE COL3

COL1:

LDI R30, LOW(KCODE0<<1)

LDI R31, HIGH(KCODE0<<1)

RJMP Find

COL2:

LDI R30, LOW(KCODE1<<1)

LDI R31, HIGH(KCODE1<<1)

RJMP Find

COL3:

LDI R30, LOW(KCODE2<<1)

LDI R31, HIGH(KCODE2<<1)

RJMP Find

Find:

LSR R21

BRCC Match

LPM R20, Z+

RJMP Find

MATCH:

RET

GROUNDCOL:

;GroundAllColumns:

LDI R20, 0x0F

OUT KEY\_PORT, R20

RET

ERASELCD:

CPI R19,0X01

BRNE NOTCLEAR

LDI R16,0X01

LDI R19,0X00

CALL CMNDWRT

NOTCLEAR:

RET

LCD\_Init:

LDI R21,0xFF

OUT LCD\_DDR, R21

LDI R16, 0x33

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x32

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x28

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x0E

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x01

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x06

CALL CMNDWRT

RET

CMNDWRT:

MOV R27, R16

ANDI R27, 0xF0

IN R26, LCD\_PRT

ANDI R26, 0x0F

OR R26, R27

OUT LCD\_PRT, R26

CBI LCD\_PRT, LCD\_RS

CBI LCD\_PRT, LCD\_RW

SBI LCD\_PRT, LCD\_EN

CALL SDELAY

CBI LCD\_PRT, LCD\_EN

CALL DELAY\_2ms

MOV R27, R16

SWAP R27

ANDI R27, 0xF0

IN R26, LCD\_PRT

ANDI R26, 0x0F

OR R26, R27

OUT LCD\_PRT, R26

SBI LCD\_PRT, LCD\_EN

CALL SDELAY

CBI LCD\_PRT, LCD\_EN

CALL DELAY\_2ms

RET

DATAWRT:

MOV R27, R16

ANDI R27, 0xF0

IN R26, LCD\_PRT

ANDI R26, 0x0F

OR R26, R27

OUT LCD\_PRT, R26

SBI LCD\_PRT, LCD\_RS

CBI LCD\_PRT, LCD\_RW

SBI LCD\_PRT, LCD\_EN

CALL SDELAY

CBI LCD\_PRT, LCD\_EN

CALL DELAY\_2ms

MOV R27, R16

SWAP R27

ANDI R27, 0xF0

IN R26, LCD\_PRT

ANDI R26, 0x0F

OR R26, R27

OUT LCD\_PRT, R26

SBI LCD\_PRT, LCD\_EN

CALL SDELAY

CBI LCD\_PRT, LCD\_EN

CALL DELAY\_2ms

RET

DELAY\_2ms:

PUSH R16

PUSH R17

LDI R16, 100

LOOP1: LDI R17,100

LOOP0: DEC R17

BRNE LOOP0

DEC R16

BRNE LOOP1

POP R17

POP R16

RET

SDELAY: NOP

NOP

NOP

NOP

NOP

NOP

RET

DELAY:

LDI R20,0XFC

LDI R21,0XF3

OUT TCNT1H,R20

OUT TCNT1L,R21

LDI R20,5

LDI R21,0X00

OUT TCCR1A,R21

OUT TCCR1B,R20

AGAIN22:

IN R20,TIFR

SBRS R20,TOV1

JMP AGAIN22

LDI R20,0X00

OUT TCCR1A, R20

OUT TCCR1B,R20

LDI R20,0X04

OUT TIFR,R20

RET

RUNMOTOR:

LDI R17,0X09 /\* INIAL PHASE \*/

LDI R19,0X00

L12:

CPI R19,0X20

BRNE GO

RET

GO:

OUT PORTB,R17

LSR R17 /\*SHIFT FOR THE NEXT SQUENCE\*/

BRCC L22 /\* IF THE CARRY IS 0 JMP TO L2\*/

ORI R17,0X8/\* LOGICAL OR WITH THE IMEDIAT VALUE TO GET THE INTIAL OR FIRST PHASE\*/

INC R19

L22:

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

CALL DELAY

JMP L12

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* serIAL COMMUNICATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SERIALCOM\_init:

/\*RECEIVING BITS SERIALLY WITH COMPLETLY INTRUPT BASED\*/

/\*STEP 1 SET THE UCSRB\*/

LDI R16,(1<<RXEN)|(1<<RXCIE)

OUT UCSRB,R16

/\*STEP 2 SET THE UCSRC REGISTER\*/

LDI R16,(1<<UCSZ1)|(1<<UCSZ0)|(1<<URSEL)

OUT UCSRC,R16

SEI

/\* STEP 3 SETING THE BAUDRATE\*/

LDI R16,0X33

OUT UBRRL,R16

RET

RXISR:

AGAIN23:

SBIS UCSRA,RXC/\* NEW DATA RECEIVED\*/

JMP AGAIN23

IN R16,UDR

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*CHECKING FOR THE DISPLAY OF TEMPERATURE\*\*\*\*\*\*\*\*\*\*\*\*\*/

CPI R16,'S'

BRNE NOTTEMPCHK

LDI R30,0

SBI PORTB,5

RETI

NOTTEMPCHK:

CPI R16,'N'

BRNE NOTTEMPCHK2

LDI R30,'1'

LDI R16,'.'

CALL DATAWRT

LDI R16,'C'

CALL DATAWRT

RETI

NOTTEMPCHK2:

CPI R30,0

BRNE NTCHK

CALL DATAWRT

RETI

NTCHK:

CPI R16,'M'

BRNE NTMOTOR

CALL RUNMOTOR

RETI

NTMOTOR:

CPI R16,'R'

BRNE NTOFFALM

CBI PORTB,6

RETI

NTOFFALM:

RETI

TRANSMIT:

SBIS UCSRA,UDRE

JMP TRANSMIT

OUT UDR,R16

RET

SERIALCOM\_init1:

/\*RECEIVING BITS SERIALLY WITH COMPLETLY INTRUPT BASED\*/

/\*STEP 1 SET THE UCSRB\*/

LDI R16,(1<<RXEN)|(1<<RXCIE)|(1<<TXEN)

OUT UCSRB,R16

/\*STEP 2 SET THE UCSRC REGISTER\*/

LDI R16,(1<<UCSZ1)|(1<<UCSZ0)|(1<<URSEL)

OUT UCSRC,R16

SEI

/\* STEP 3 SETING THE BAUDRATE\*/

LDI R16,0X33

OUT UBRRL,R16

RET

CODE OF 2 ATMEGA

.ORG 0X00

JMP MAIN

.ORG URXCaddr

JMP RXISR

.ORG ADCCaddr

JMP ADCISR

MAIN:

/\*\*\*\*\*\*\*\*\*++++++ PORTD FOR THE OUTPUT OF BLUETOOTH TO CONTROL THE APPLIANCES\*\*\*\* ++++R20-------------\*/

LDI R20,0XFF

OUT DDRC,R20

/\*---R20\*\*\*\*\*+++++++++++++++++++ PORTA AND PORTB ARE USING IN LCD INTIALIZATION++++++++++++++++\*/

.equ LCD\_PRT = PORTB

.equ LCD\_DDR = DDRB

.equ LCD\_PIN = PINB

.equ LCD\_RS = 0

.equ LCD\_RW = 1

.equ LCD\_EN = 2

/\*\*\*\*\*\*\*\*\*\*\*\*\*STACK INTIALIZATION\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R21,HIGH(RAMEND)

OUT SPH,R21

LDI R21,LOW(RAMEND)

OUT SPL,R21

SBI PORTC,0

SBI PORTC,3

CALL SERIALCOM\_init

call LCD\_Init

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DOING THE MAIN TASKS\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R16, 'F'

CALL DATAWRT

LDI R16, 'A'

CALL DATAWRT

LDI R16, 'S'

CALL DATAWRT

LDI R16,'T'

CALL DATAWRT

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* FOR SENSOR\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R16,0X00

OUT PORTA,R16/\* AS THE CHANNEL IS ALL FROM PORTA THEN MAKE ALL OF IS PINS TO INPUT PIN\*/

LDI R16,0X8F/\* ENABLING THE ADC AND STTING THE PRESCALAR CK/128 AND SETTING UP THE INTRUPT 1000 1111\*/

OUT ADCSRA,R16

LDI R16,225 /\*PUTTING THE BUILDIN VREF from the ref pin AND PRESCALAR OF CLK/1024\*/

OUT ADMUX,R16

A1:

JMP A1

LCD\_Init:

LDI R21,0xFF

OUT LCD\_DDR, R21

LDI R16, 0x33

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x32

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x28

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x0E

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x01

CALL CMNDWRT

CALL DELAY\_2ms

LDI R16, 0x06

CALL CMNDWRT

RET

CMNDWRT:

MOV R27, R16

ANDI R27, 0xF0

IN R26, LCD\_PRT

ANDI R26, 0x0F

OR R26, R27

OUT LCD\_PRT, R26

CBI LCD\_PRT, LCD\_RS

CBI LCD\_PRT, LCD\_RW

SBI LCD\_PRT, LCD\_EN

CALL SDELAY

CBI LCD\_PRT, LCD\_EN

CALL DELAY\_2ms

MOV R27, R16

SWAP R27

ANDI R27, 0xF0

IN R26, LCD\_PRT

ANDI R26, 0x0F

OR R26, R27

OUT LCD\_PRT, R26

SBI LCD\_PRT, LCD\_EN

CALL SDELAY

CBI LCD\_PRT, LCD\_EN

CALL DELAY\_2ms

RET

DATAWRT:

MOV R27, R16

ANDI R27, 0xF0

IN R26, LCD\_PRT

ANDI R26, 0x0F

OR R26, R27

OUT LCD\_PRT, R26

SBI LCD\_PRT, LCD\_RS

CBI LCD\_PRT, LCD\_RW

SBI LCD\_PRT, LCD\_EN

CALL SDELAY

CBI LCD\_PRT, LCD\_EN

CALL DELAY\_2ms

MOV R27, R16

SWAP R27

ANDI R27, 0xF0

IN R26, LCD\_PRT

ANDI R26, 0x0F

OR R26, R27

OUT LCD\_PRT, R26

SBI LCD\_PRT, LCD\_EN

CALL SDELAY

CBI LCD\_PRT, LCD\_EN

CALL DELAY\_2ms

RET

DELAY\_2ms:

PUSH R16

PUSH R17

LDI R16, 100

LOOP1: LDI R17,100

LOOP0: DEC R17

BRNE LOOP0

DEC R16

BRNE LOOP1

POP R17

POP R16

RET

SDELAY: NOP

NOP

NOP

NOP

NOP

NOP

RET

RXISR:

AGAIN:

SBIS UCSRA,RXC/\* NEW DATA RECEIVED \*/

JMP AGAIN

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*++++++++++++++R20\*/

IN R16,UDR

CALL DATAWRT

/\*\*\*\*\*\*\*\*\*\* CHECKING FOR TEMPERATURE\*\*\*\*\*\*\*\*/

LDI R20,'T'

CP R16,R20

BRNE NTTEMP

LDI R16,'S'

CALL TRANSMIT

SBI ADCSRA,ADSC/\* ENABLE THE CONVERSION OR START CONVERSION\*/

RETI

NTTEMP:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CHECKING FOR 0\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R20,'0'

CP R16,R20

BRNE CP0

SBI PORTC,0

RETI

CP0:

LDI R20,'1'

CP R16,R20

BRNE CP1

SBI PORTC,1

LDI R16,'M'

CALL TRANSMIT

RETI

CP1:

LDI R20,'2'

CP R16,R20

BRNE CP2

SBI PORTC,2

LDI R16,'R'

CALL TRANSMIT

RETI

CP2:

LDI R20,'3'

CP R16,R20

BRNE CP3

SBI PORTC,3

RETI

CP3:

LDI R20,'4'

CP R16,R20

BRNE CP4

CBI PORTC,0

RETI

CP4:

LDI R20,'5'

CP R16,R20

BRNE CP5

CBI PORTC,1

RETI

CP5:

LDI R20,'6'

CP R16,R20

BRNE CP6

CBI PORTC,2

RETI

CP6:

LDI R20,'7'

CP R16,R20

BRNE CP7

CBI PORTC,3

RETI

CP7:

LDI R20,'8'

CP R16,R20

BRNE CP8

CBI PORTC,0

RETI

CP8:

RETI

DELAY:

LDI R20,0XFC

LDI R21,0XF3

OUT TCNT1H,R20

OUT TCNT1L,R21

LDI R20,5

LDI R21,0X00

OUT TCCR1A,R21

OUT TCCR1B,R20

AGAIN22:

IN R20,TIFR

SBRS R20,TOV1

JMP AGAIN22

LDI R20,0X00

OUT TCCR1A, R20

OUT TCCR1B,R20

LDI R20,0X04

OUT TIFR,R20

RET

ADCISR:

SBI ADCSRA,ADIF/\* RESET THE FLAG FOR THE NEXT CONVERSION\*/

IN R16,ADCH

CALL CVTHEX /\* FUNCTION THAT TAKE THE INPUT CONVERT TO THE FORM TO DISPLAY ON LCD\*/

LDI R16,'.'

CALL DATAWRT

LDI R16,'C'

CALL DATAWRT

RETI

CVTHEX:

/\* STORING R16\*\*\*/

MOV R18,R16

/\*\*\*\*\*\*\*\*\*\*\*\*\*CHECKING FOR LESS THAN 10\*\*\*\*\*\*\*\*\*\*/

LDI R17,10

CP R16,R17

BRSH CON

CALL FINDNUMBER

LPM R16,Z

CALL DATAWRT

RET

CON:

/\*\*\*\*\*\*\*\*\*\*\* CHECKING FOR LESS THAN 19\*\*\*\*\*\*\*\*\*\*\*/

LDI R17,20

CP R16,R17

BRSH CON1

LDI R16,'1'

CALL TRANSMIT

CALL DATAWRT

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*CODE FOR LESS THAN 19\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SUBI R18,10

CPI R18,0

BRNE NOTZERO

LDI R16,'0'

CALL TRANSMIT

CALL DATAWRT

LDI R16,'N'

CALL TRANSMIT

RET

NOTZERO:

CALL FINDNUMBER

LPM R16,Z

CALL DATAWRT

RET

CON1:

/\*\*\*\*\*\*\*\*\*\*\*\* CHECKING FOR LESS THAN 29\*\*\*\*\*\*\*\*\*/

LDI R17,30

CP R16,R17

BRSH CON2

LDI R16,'2'

CALL TRANSMIT

CALL DATAWRT

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CODE FOR LESS THAN 29\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

SUBI R18,20

CPI R18,0

BRNE NOTZERO1

LDI R16,'0'

CALL TRANSMIT

CALL DATAWRT

LDI R16,'N'

CALL TRANSMIT

RET

NOTZERO1:

CALL FINDNUMBER

LPM R16,Z

CALL DATAWRT

RET

CON2:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CHECKING FOR LESS THAN 39\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R17,40

CP R16,R17

BRSH CON3

LDI R16,'3'

CALL TRANSMIT

CALL DATAWRT

SUBI R18,30

CPI R18,0

BRNE NOTZERO2

LDI R16,'0'

CALL TRANSMIT

CALL DATAWRT

LDI R16,'N'

CALL TRANSMIT

RET

NOTZERO2:

CALL FINDNUMBER

LPM R16,Z

CALL DATAWRT

RET

CON3:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CHECKING FOR LESS THAN 49\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R17,50

CP R16,R17

BRSH CON4

LDI R16,'4'

CALL TRANSMIT

CALL DATAWRT

SUBI R18,40

CPI R18,0

BRNE NOTZERO3

LDI R16,'0'

CALL TRANSMIT

CALL DATAWRT

LDI R16,'N'

CALL TRANSMIT

RET

NOTZERO3:

CALL FINDNUMBER

LPM R16,Z

CALL DATAWRT

RET

CON4:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*CHECKING FOR LESS THAN 59\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

LDI R17,60

CP R16,R17

BRSH CON5

LDI R16,'5'

CALL TRANSMIT

CALL DATAWRT

SUBI R18,50

CPI R18,0

BRNE NOTZERO4

LDI R16,'0'

CALL TRANSMIT

CALL DATAWRT

LDI R16,'N'

CALL TRANSMIT

RET

NOTZERO4:

CALL FINDNUMBER

LPM R16,Z

CALL DATAWRT

RET

CON5:

RET

FINDNUMBER:

LDI ZH,HIGH(NUMBERS<<1)

LDI ZL,LOW(NUMBERS<<1)

LDI R16,1

AG:

CP R18,R16

BRNE AG1

LPM R16,Z

CALL TRANSMIT

LDI R16,'N'

CALL TRANSMIT

RET

AG1:

INC ZL

DEC R18

JMP AG

NUMBERS:

.DB '1','2','3','4','5','6','7','8','9'

TRANSMIT:

SBIS UCSRA,UDRE

JMP TRANSMIT

OUT UDR,R16

RET

SERIALCOM\_init:

/\*RECEIVING BITS SERIALLY WITH COMPLETLY INTRUPT BASED\*/

/\*STEP 1 SET THE UCSRB\*/

LDI R16,(1<<RXEN)|(1<<RXCIE)|(1<<TXEN)

OUT UCSRB,R16

/\*STEP 2 SET THE UCSRC REGISTER\*/

LDI R16,(1<<UCSZ1)|(1<<UCSZ0)|(1<<URSEL)

OUT UCSRC,R16

SEI

/\* STEP 3 SETING THE BAUDRATE\*/

LDI R16,0X33

OUT UBRRL,R16

RET