

Switcher

Ashley J. Robinson

January 4, 2017

1 Introduction

The task is to design and build a switch mode power supply (SMPS) which will operate in boost and buck mode. The controller will contain a selection of menu options allowing the user to control as many features as possible of the SMPS operation. The software running on the controlling processor must consist of no more than 1024 bytes of machine code.

1.1 Processor selection

The chosen processor is a **Silicon Labs C8051F850-C-GU** which uses the 8051 architecture. An 8051 processor is a good choice for a code size reduction exercise as there are many thoroughly tested compilers available. The Keil c51 compiler will be used to generate assembly for this project.

2 Hardware

A soldering iron and solder is required to build the switcher hardware. The schematic is held in Figure 1 along with the BoM in Table 1. Photos of the top and bottom of the stripboard build are contained in Figures 2 and 3. Note the visible modification to the development board is not required to replicate the design however **R5, R6 and R9 must be removed from the board development board.**

2.1 Schematic

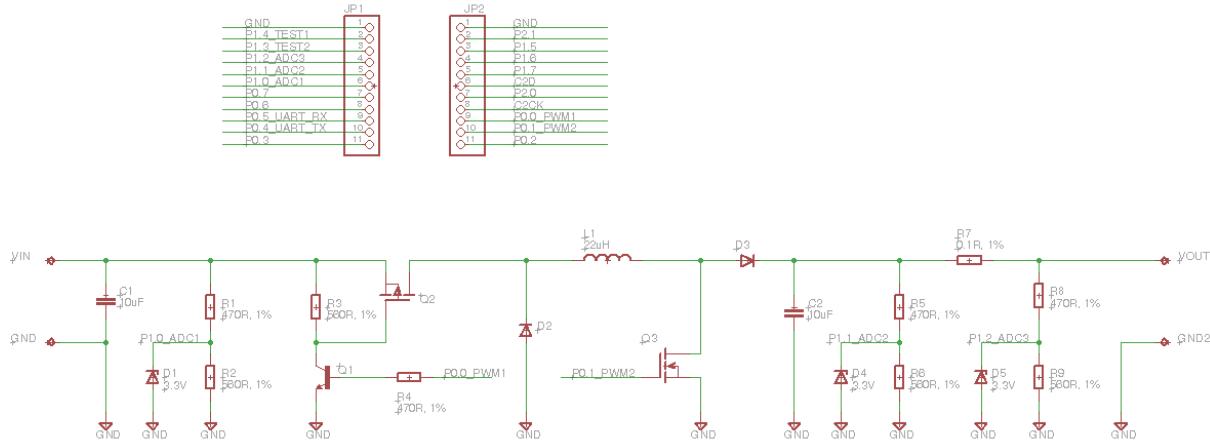


Figure 1: Schematic.

2.2 Bill of Materials (BoM)

Designator	Value	Manufacturer	Manufacturer Number	Part Number
C1,C2	10uF	MULTICOMP		MCGPR25V106M5X11
D1,D4,D5	3.3V	ON SEMICONDUCTOR		1N5333BG
D2,D3		TAIWAN SEMICONDUCTOR		SR1504
JP1,JP2		AMPHENOL FCI		77311-401-36LF
L1	22uH	PANASONIC		ELC08D220E
Q1		MULTICOMP		BC337
Q2		FAIRCHILD SEMICONDUCTOR		FQP27P06
Q3		FAIRCHILD SEMICONDUCTOR		FQP30N06L
R1,R4,R5,R8	470Ω	MULTICOMP		MF12 470R
R2,R3,R6,R9	560Ω	MULTICOMP		MF12 560R
R7	0.1Ω	VISHAY		LVR01R1000FE12
Strip board		KEMO		ELECTRONIC E012
Dev board		SILICON LABS		LTOOLSTICK850DC-UG

Table 1: BoM

2.3 Stripboard

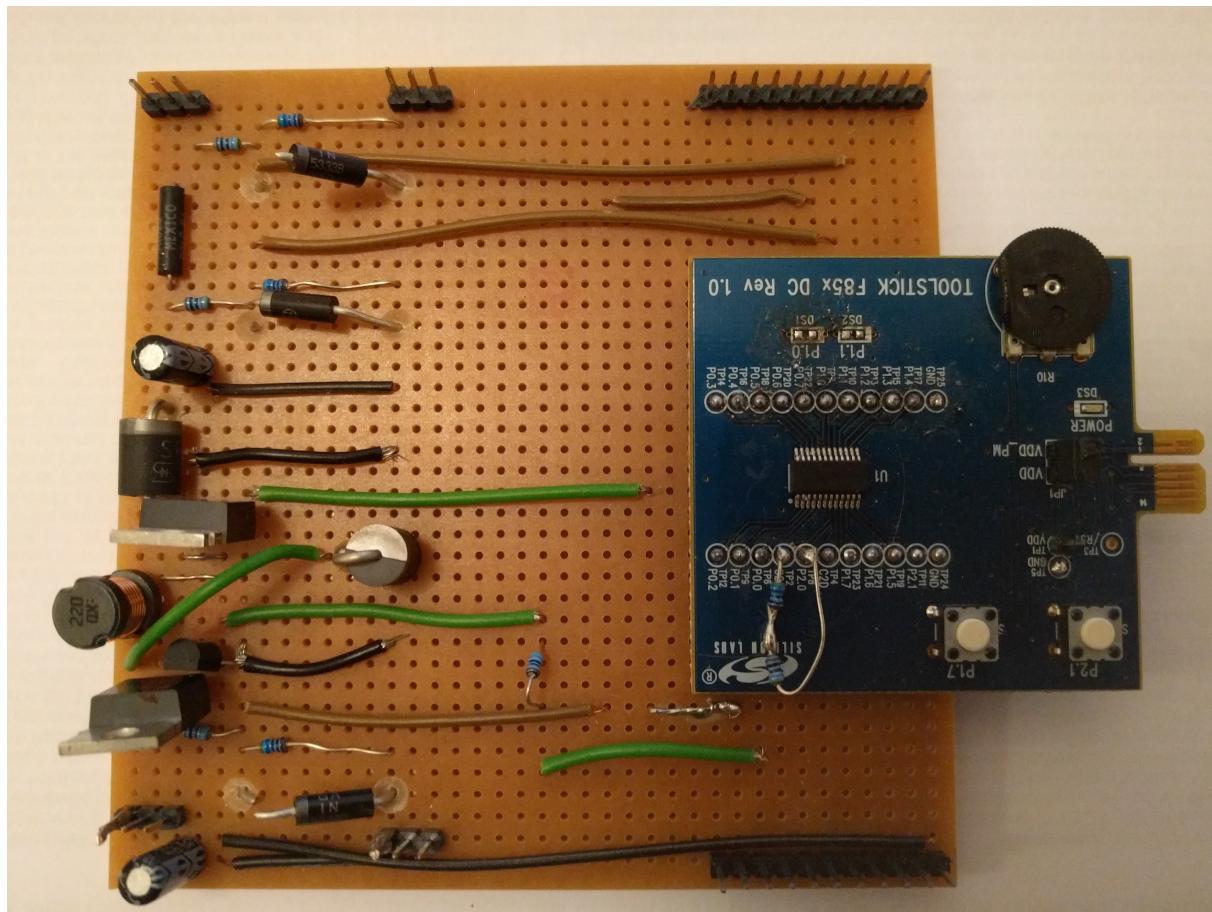


Figure 2: Stripboard top side.

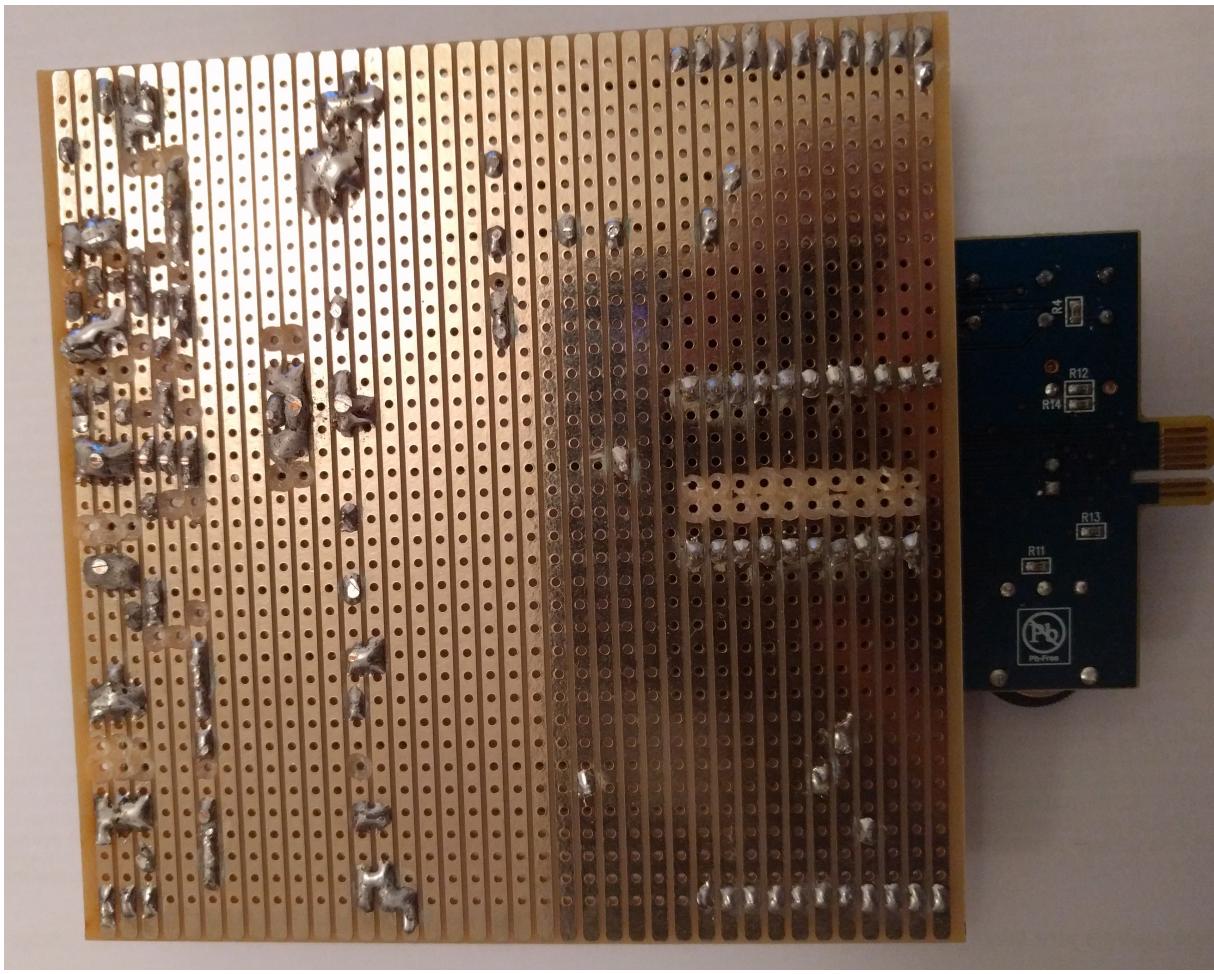


Figure 3: Stripboard bottom side.

3 Software

3.1 Peripherals

3.1.1 Analogue to Digital Convertors (ADCs)

A single ADC is multiplexed to measure three voltages on the board.

3.1.2 Universal Asynchronouse/Syncronous Transceiver (UART)

Transmission from the microcontroller happens through `uartLoadOut` which adds to the **8 byte** buffer and is then unloaded from a timer. The input is not interrupt driven and is handled in the main loop using a **5 byte** buffer that is enough to contain the longest command.

3.1.3 Programmable Counter Array (PCA)

The Pulse Width Modulation (PWM) is controlled from the counter array. The output runs at approximatly 96KHz with 8 bits to control the duty cycle. A high resolution for control would be favourable for this application but the frequency achieved in 16-bit mode is far too low for this application.

3.1.4 Timers

Timer 0 is used as baud rate generation for the UART. **Timer 2** is used to trigger an Interrupt Service Routine (ISR) which runs at 4KHz. The ISR controls the UART transmission, sampling of the ADC, running the controller and finally setting the PWM.

3.2 Operation

A brief description of the software operation is contained in Figure 4 and the full code in Listing 3.

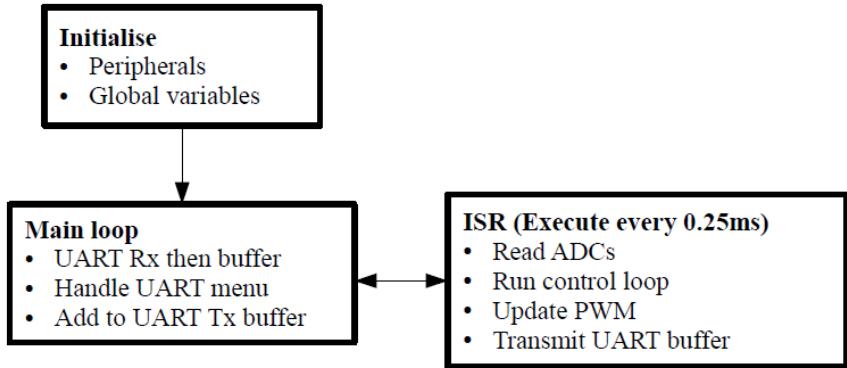


Figure 4: Software overview.

3.3 UART Menu

The baud rate of the UART is 115200 with 8 data bits, no parity or control flow. The commands in Table 2 reference the single characters for the menu and amount of the data to be exchanged.

Command	Char	Send numbers	Return numbers	Notes
Enable	g	0	0	0
Disable	s	0	0	0
Read ADC1	x	0	4	Result in mV
Read ADC2	y	0	4	Result in mV
Read ADC3	z	0	4	Result in mV
Read PWM duty	d	0	4	Percent = Result /2.55
Read output current	c	0	4	Result in mA
Set output voltage	v	4	0	Send value in mV
Set input voltage upper limit	h	4	0	Send value in mV
Set input voltage lower limit	l	4	0	Send value in mV
Read set output voltage	a	0	4	Result value in mV
Read set input voltage upper limit	m	0	4	Result in mV
Read set input voltage lower limit	n	0	4	Result in mV

Table 2: UART menu

3.4 Assembly analysis

The following sections of code are library functions inserted by the compiler implicitly to facilitate some more complex operations. A total of 146 lines of assembly are inserted for the implicit functions; not including any extra code to build stack frames, etc.

3.4.1 ADC scaling

Listing 1 is a line of code operating on a 32 bit signed integer and invokes listing 5 and 6. The purpose of this operation is to cast the voltage recorded by the ADC to a representation in mV. It is possible for the target voltage to be translated instead but this would still require the same piece of code. The 12 bit output but the ADC must be multiplied 5.926 as to represent the voltage at the top of the potential divider. Fixed bit multiplication followed by a shift operation removes the need for a floating point operation but yields the same result. The value of the potential dividers can be changed to improve the operation required. The additional lines of assembly total 57 from using this operation.

Listing 1: ADC

```
return (((U32)ADCO)*SCALE_MUL) >> 10;
```

3.4.2 Division

Division and modulus operations are contained in 2 invoke the functions contained in 8 which contains 62 lines of assembly.

Listing 2: Division

```
scale /= 10;
```

```

num = out / scale;
out %= scale;

```

3.4.3 Switch

Using a case statement will generate different assembly from an if/else ladder. The function in Listing 9 is used to control the switch statement and totals 27 lines of assembly.

4 Testing

The potential dividers constantly place the power supply under a load of 515Ω . To test both the UART interface and the switch mode functionality a small python script is used to create GUI to plot values in real time. The input and output voltages recorded by the microcontroller are contained in Figure 5 and the actual values are contained in 6.

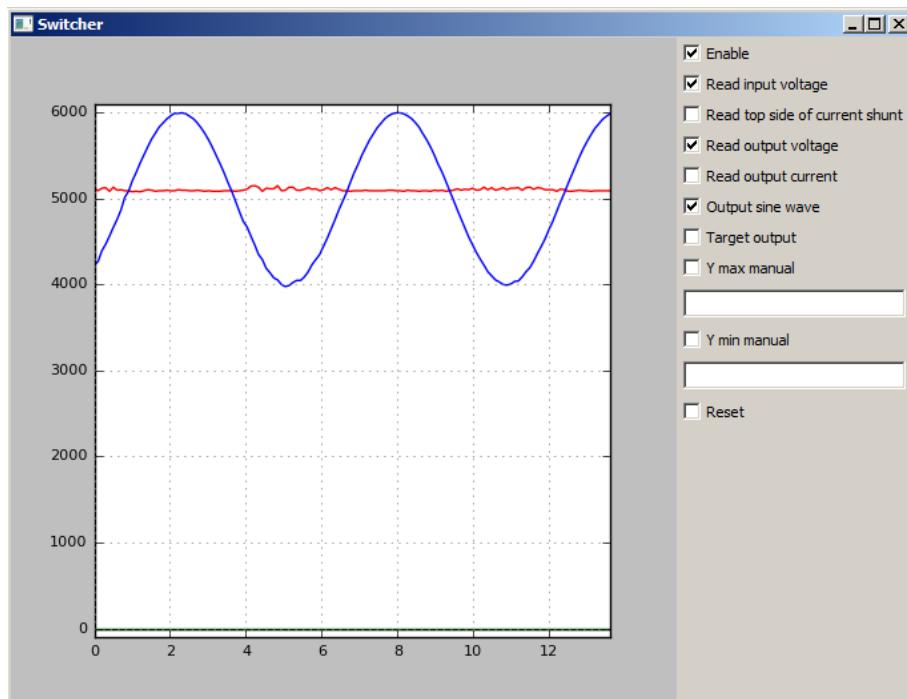


Figure 5: Sine wave output on GUI.

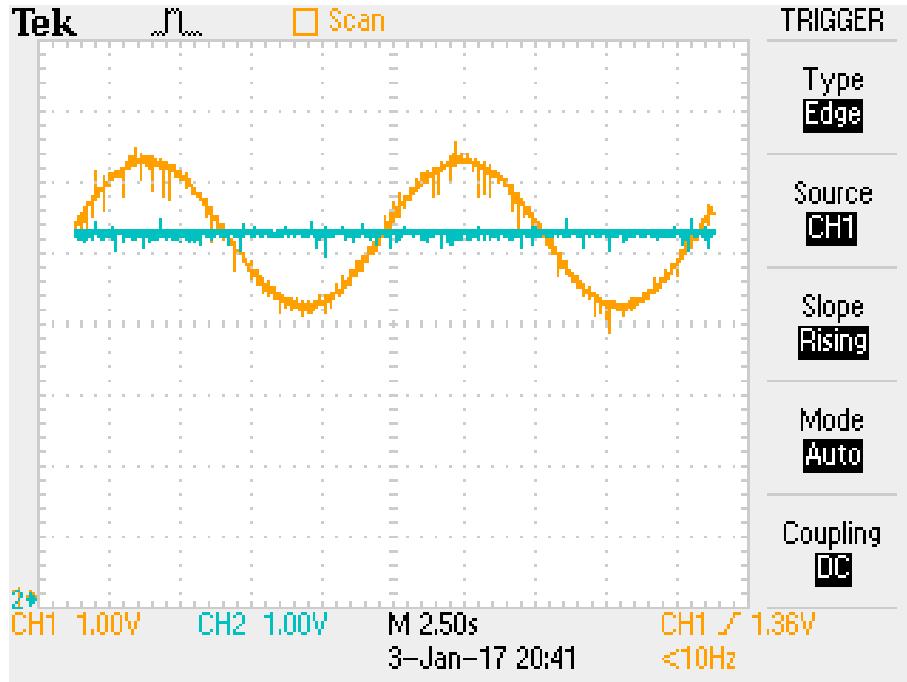


Figure 6: Sine wave output on scope.

5 Conclusion

The complete assembly in listing 4 is generated from the code in listing 3. The C code is contained within a single file and generates 950 lines of assembler with an additional 3 lines contained within the jump table and the jump on reset...

- **0x0000** Jumps straight to 0x0800
- **0x001B** Jump table entry TIMER1_ISR(C:0BB6)
- **0x002B** Jump table entry TIMER2_ISR(C:094C)
- **0x0800** First line of code generated from main.c
- **0x0BB6** Final line of code generated from main.c

The Keil c51 compilier limits machine code generation to 2KB (not a problem) however it also limits the use of memory less than 2KB by always jumping to address 2048 and then continuing with the compiled assembler. Excluding the first 2048 is not completely fair because the jump on reset and the jump table would still exsist. The highest vector used in the jump table is at address 43 so assuming this is not optimised by the compilier the total code size comes to **993 bytes**.

5.1 Code size over time

The graph in 7 pltos code size agaiinst discret commits. Initial development is erratic, the size slowly grows as more features are added and eventually the code is optimised back below the limit.

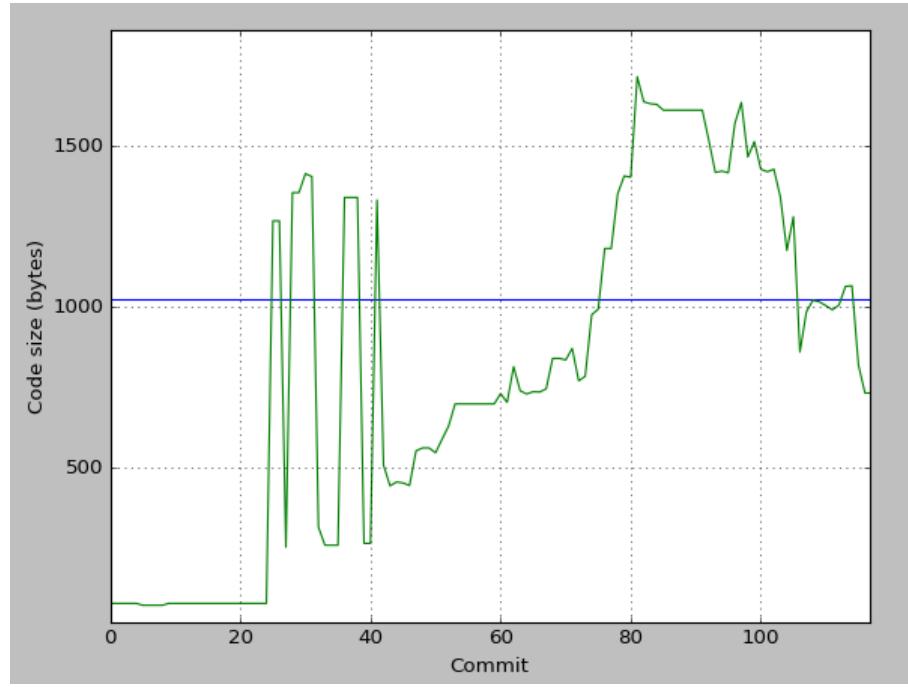


Figure 7: Code size against commits.

5.2 Further work

- Improve output current
- Replaced protection diodes with suitable parts
- Improve resolution of ADC to enable constant control

6 Appendix

6.1 Code listings

Listing 3: main.c

```

//-----
// Project: Swticher
// File:    main.c
// Brief:   Single main file containing all code
//-----

//-----
// Includes
//-----

#include "SI_C8051F850_Register.Enums.h"
#include "SI_C8051F850_Defs.h"

//-----
// Defines
//-----

#define SYSCLK          24500000          // SYSCLK frequency in Hz
#define BAUDRATE        115200           // Baud rate of UART in bps

#define P               5
#define I               2

#define UART_SIZE_IN   5
#define UART_SIZE_OUT  8

#define ADC1            0x08
#define ADC2            0x09
#define ADC3            0x0A

```

```

#define SCALE_MUL          6068           // When combined with >> 10 will scale by 5.926 to compensate for potential divide

#define DEFAULT_OUT_MV    5000
#define DEFAULT_HIGH_MV   6000
#define DEFAULT_LOW_MV    4000

SBIT(TEST2, SFR_P1, 3);           // DS5 P1.0 LED
SBIT(TEST1, SFR_P1, 4);           // DS5 P1.0 LED

//-----
// Prototypes
//-----

U16 readAdc(U8 sel);
void uartLoadOut(U8 tx);
U16 uartNumbers(U16 toSend, bool transmit);

//-----
// Global Variables
//-----

volatile U8      uart_in[UART_SIZE_IN];
volatile U8      uart_out[UART_SIZE_OUT];
volatile U8      head;
volatile U8      tail;
volatile int     integral;
volatile U8      duty;
volatile U16    high_mV;
volatile U16    low_mV;
volatile U16    target_mV;
volatile bool   enabled;
volatile U16    adc1;
volatile U16    adc2;
volatile U16    adc3;
volatile U16    current;

//-----
// Main Routine
//-----

void main (void){
    // Start of peripheral setup
    U8 TCON_save;
    // Watchdog
    WDTCN = 0xDE;                      // First key
    WDTCN = 0xAD;                      // Second key - Watchdog now disabled
    // Clock
    CLKSEL =
        CLKSEL_CLKSL__HFOSC            | // Use 24.5MHz internal clock
        CLKSEL_CLKDIV__SYSCLK_DIV_1;   | // Do not divide
    // Port 0
    P0MDOUT =
        P0MDOUT_BO__PUSH_PULL          | // PWM1 output
        P0MDOUT_B1__PUSH_PULL          | // PWM2 output
        P0MDOUT_B2__OPEN_DRAIN         |
        P0MDOUT_B3__OPEN_DRAIN         |
        P0MDOUT_B4__PUSH_PULL          | // UART TX
        P0MDOUT_B5__OPEN_DRAIN         | // UART RX
        P0MDOUT_B6__OPEN_DRAIN         |
        P0MDOUT_B7__OPEN_DRAIN         ;
    // Port 1
    P1MDOUT =
        P1MDOUT_BO__OPEN_DRAIN          |
        P1MDOUT_B1__OPEN_DRAIN          |
        P1MDOUT_B2__OPEN_DRAIN          |
        P1MDOUT_B3__PUSH_PULL          | // TEST2
        P1MDOUT_B4__PUSH_PULL          | // TEST1
        P1MDOUT_B5__OPEN_DRAIN          |
        P1MDOUT_B6__OPEN_DRAIN          |
        P1MDOUT_B7__OPEN_DRAIN          ;
    P1MDIN =
        P1MDIN_BO__ANALOG              | // ADC1
        P1MDIN_B1__ANALOG              | // ADC2
        P1MDIN_B2__ANALOG              | // ADC3
        P1MDIN_B3__DIGITAL             |
        P1MDIN_B4__DIGITAL             |
        P1MDIN_B5__DIGITAL             |
        P1MDIN_B6__DIGITAL             |
        P1MDIN_B7__DIGITAL             ;
    // Port crossbar
    XBR0 =
        XBRO_UTRTOE__ENABLED           | // Route out UART
        XBRO_SPIOE__DISABLED           |
        XBRO_SNBOE__DISABLED           |
        XBRO_CPOE__DISABLED           |
        XBRO_CPOAE__DISABLED           |
        XBRO_CP1IE__DISABLED           |
        XBRO_CP1AE__DISABLED           |
        XBRO_SYSCKE__DISABLED           ;
    XBR1 =
        XBR1_PCAOME__CEXO_CEX1         | // Route out PCA0 and PCA1
        XBR1_ECIE__DISABLED           |
        XBR1_TOE__DISABLED             |
        XBR1_TIE__DISABLED             |
        XBR1_T2E__DISABLED             ;
    XBR2 =
        XBR2_WEAKPUD__PULL_UPS_ENABLED | // Weak pull ups
        XBR2_XBARE__ENABLED           | // Enable cross bar
}

```

```

// ADC
//ADCOMX =
//    ADCOMX_ADCOMX_ADCOP10;
ADCOCF =
(1 << ADCOCF_ADSC__SHIFT) | // Mux set in application
ADCOCF_ADS8E__NORMAL | // ADC set to 10 bit
ADCOCF_ADGN__GAIN_1 | // ADC gain set to 1
ADCOCF_ATDM__TRACK_NORMAL; // Immediate convert
ADCOCNO &=
~ADCOCNO_ADCM__FMASK;
ADCOCNO |=
ADCOCNO_ADEN__ENABLED
ADCOCNO_ADMC__ABUSY;

// Timer 0
TCON_save = TCON;
TCON &=
TCON_TRO__BMASK
TCON_TR1__BMASK;
TH1 = (150 << TH1_TH1__SHIFT);
TL1 = (150 << TL1_TL1__SHIFT);
TCON =
TCON_save;

// Timer setup
CKCON =
CKCON_SCA__SYSCLK_DIV_12
CKCON_TOM__PRESCALE
CKCON_T3MH__EXTERNAL_CLOCK
CKCON_T3ML__EXTERNAL_CLOCK
CKCON_T1M__SYSCLK;

TMOD =
TMOD_TOM__MODE0
TMOD_CTO__TIMER
TMOD_GATE0__DISABLED
TMOD_TIM__MODE2
TMOD_CT1__TIMER
TMOD_GATE1__DISABLED;
TCON |=
TCON_TR1__RUN ;

// UART
SCONO |=
SCONO_REN__RECEIVE_ENABLED;

// VREF
REFOCN =
REFOCN_REFSL__VDD_PIN
REFOCN_IREFLVL__1P65
REFOCN_GNDSL__GND_PIN
REFOCN_TEMPE__TEMP_DISABLED;

// RSTSRC
RSTSRC =
RSTSRC_COSEF__NOT_SET
RSTSRC_MCDRSF__SET
RSTSRC_PORSF__SET
RSTSRC_SWRSF__NOT_SET;

// Interrupt
IE =
IE_EA__ENABLED
IE_EX0__DISABLED
IE_EX1__DISABLED
IE_ESPIO__DISABLED
IE_ETO__DISABLED
IE_ET1__ENABLED
IE_ET2__ENABLED
IE_ESO__ENABLED;

IP =
IP_PX0__LOW
IP_PX1__LOW
IP_PSPIO__LOW
IP_PTO__LOW
IP_PT1__LOW
IP_PT2__HIGH
IP_PSO__LOW; // Timer 2 highest priority

// PCA
PCAOCN_CR =
PCAOCN_CR__STOP;
PCAOPMO =
PCAOPMO_CAPN__DISABLED
PCAOPMO_ECCF__ENABLED
PCAOPMO_MAT__ENABLED
PCAOPMO_CAPP__DISABLED
PCAOPMO_ECOM__ENABLED
PCAOPMO_PWM__ENABLED
PCAOPMO_TOG__DISABLED;
PCAOPM1 =
PCAOPM1_CAPN__DISABLED
PCAOPM1_ECCF__ENABLED
PCAOPM1_MAT__ENABLED
PCAOPM1_CAPP__DISABLED
PCAOPM1_ECOM__ENABLED
PCAOPM1_PWM__ENABLED
PCAOPM1_TOG__DISABLED;
PCAOMD =
PCAOMD_CIDL__NORMAL
PCAOMD_ECF__OVF_INT_DISABLED
PCAOMD_CPS__SYSCLK;
PCAOCN |=
PCAOCN_CR__RUN;

// Timer 2
TMR2CN |= TMR2CN_TR2__RUN;
// End of peripheral setup

```

```

enabled      = false;
integral     = 0;
target_mV   = DEFAULT_OUT_MV;
high_mV     = DEFAULT_HIGH_MV;
low_mV      = DEFAULT_LOW_MV;

SCONO_TI    = 1;
SCONO_RI    = 0;

while (1){
    if(SCONO_RI){
        SCONO_RI = 0;
        uart_in[4] = uart_in[3];
        uart_in[3] = uart_in[2];
        uart_in[2] = uart_in[1];
        uart_in[1] = uart_in[0];
        uart_in[0] = SBUFO;
        // Small buffer, less code not to use loop
        uartLoadOut(uart_in[0]);

        switch(uart_in[0]){
            case 'a': uartNumbers(target_mV,true);
            break;
            case 'c': uartNumbers(current,true);
            break;
            case 'd': uartNumbers(duty,true);
            break;
            case 'g': enabled      = true;
            integral     = 0;
            break;
            case 's': enabled      = false;
            break;
            case 'm': uartNumbers(high_mV,true);
            break;
            case 'n': uartNumbers(low_mV,true);
            break;
            case 'x': uartNumbers(adcl,true);
            break;
            case 'y': uartNumbers(adc2,true);
            break;
            case 'z': uartNumbers(adc3,true);
            break;
            default: switch(uart_in[4]){
                case 'h': high_mV      = uartNumbers(high_mV,false);
                break;
                case 'l': low_mV       = uartNumbers(low_mV,false);
                break;
                case 'v': target_mV   = uartNumbers(target_mV,false);
                integral     = 0;
                break;
            }
            break;
        }
    }
}
}

//-----
// Routines
//-----
void uartLoadOut(U8 tx){                                // Handle buffering out Tx UART
    uart_out[head] = tx;                               // Buffer outgoing
    head++;
    head %= UART_SIZE_OUT;                           // Wrap around
}

U16 uartNumbers(U16 toSend, bool transmit){           // Tx/Rx up to 4 length numbers over UART
    U16 out = toSend;
    U16 num = 0;
    U16 scale = 10000;
    U8 test;
    U8 i = 4;
    bool bad = false;
    while(i){
        scale /= 10;                                // On zero done
        i--;
        test = uart_in[i] - 48;                      // Shift
        // Move through UART array
        if(test > 10){                             // ascii to num
            bad = true;                            // check is 0 to 9, unsigned
        }
        num += test*scale;                          // shift in to position
        if(transmit){                            // Put if statement at back of loop to save on jumps
            num = out / scale;                    // 10 powers
            uartLoadOut(num + 48);               // Number to ascii
            out %= scale;                        // Remainder for next time
        }
    }
    uartLoadOut('\n');
    uartLoadOut('\r');
    if(bad){                                // Not all valid numbers so set output as default
        num = DEFAULT_OUT_MV;
    }
    return num;
}

U16 readAdc(U8 sel){                                // Read the available ADCs
    U8 i;

```

```

ADCOMX = sel;
for(i=0;i<2;i++){
    ADCOCNO |= ADCOCNO_ADBUSY_SET;
    while(ADCOCNO & ADCOCNO_ADBUSY_SET);           // Wait for sample to complete
}
return (((U32)ADCO)*SCALE_MUL) >> 10;           // Scale to mV
}

//-----
// Interrupt Routines
//-----

INTERRUPT (TIMER1_ISR, TIMER1_IRQn){}           // Needed for UART timing
INTERRUPT (TIMER2_ISR, TIMER2_IRQn){}           // One timer handling UART Tx and PID
int out;
int error;
TEST1 = 1;

adc1 = readAdc(ADC1);
adc2 = readAdc(ADC2);
adc3 = readAdc(ADC3);
current = (adc2 - adc3)*10;

if((adc1 < low_mV) || (adc1 > high_mV)){        // Watch input voltage
    enabled = false;
}

error = (int)target_mV - (int)adc3;               // PID controller
integral += error;
out = ((error*P) + (integral*I)) >> 10;         // Divide by 1024
if((out < 0) || (!enabled)){
    out = 0;
}
duty = out;
PCAOPH0 = PCAOPH1 = 0xFF - out;

if(head != tail){
    SBUFO = uart_out[tail];                      // Timer tuned so no need to check
    tail++;                                       // Transmit UART
    tail %= UART_SIZE_OUT;                       // Wrap around
}

TMR2H = 255;                                     // Runs at 4KHz
TMR2CN.TF2H = 0;                                 // Enable interrupt again
TEST1 = 0;
}

//-----
// END
//-----

```

Listing 4: Swicther.asm

C:0x0000	020BAA	LJMP	C:0BAA
C:0x0003	00	NOP	
C:0x0004	00	NOP	
C:0x0005	00	NOP	
C:0x0006	00	NOP	
C:0x0007	00	NOP	
C:0x0008	00	NOP	
C:0x0009	00	NOP	
C:0x000A	00	NOP	
C:0x000B	00	NOP	
C:0x000C	00	NOP	
C:0x000D	00	NOP	
C:0x000E	00	NOP	
C:0x000F	00	NOP	
C:0x0010	00	NOP	
C:0x0011	00	NOP	
C:0x0012	00	NOP	
C:0x0013	00	NOP	
C:0x0014	00	NOP	
C:0x0015	00	NOP	
C:0x0016	00	NOP	
C:0x0017	00	NOP	
C:0x0018	00	NOP	
C:0x0019	00	NOP	
C:0x001A	00	NOP	
C:0x001B	020BB6	LJMP	TIMER1_ISR(C:0BB6)
C:0x001E	00	NOP	
C:0x001F	00	NOP	
C:0x0020	00	NOP	
C:0x0021	00	NOP	
C:0x0022	00	NOP	
C:0x0023	00	NOP	
C:0x0024	00	NOP	
C:0x0025	00	NOP	
C:0x0026	00	NOP	
C:0x0027	00	NOP	
C:0x0028	00	NOP	
C:0x0029	00	NOP	

C :0x002A	00	NOP
C :0x002B	02094C	LJMP TIMER2_ISR(C:094C)
C :0x002E	00	NOP
C :0x002F	00	NOP
C :0x0030	00	NOP
C :0x0031	00	NOP
C :0x0032	00	NOP
C :0x0033	00	NOP
C :0x0034	00	NOP
C :0x0035	00	NOP
C :0x0036	00	NOP
C :0x0037	00	NOP
C :0x0038	00	NOP
C :0x0039	00	NOP
C :0x003A	00	NOP
C :0x003B	00	NOP
C :0x003C	00	NOP
C :0x003D	00	NOP
C :0x003E	00	NOP
C :0x003F	00	NOP
C :0x0040	00	NOP
C :0x0041	00	NOP
C :0x0042	00	NOP
C :0x0043	00	NOP
C :0x0044	00	NOP
C :0x0045	00	NOP
C :0x0046	00	NOP
C :0x0047	00	NOP
C :0x0048	00	NOP
C :0x0049	00	NOP
C :0x004A	00	NOP
C :0x004B	00	NOP
C :0x004C	00	NOP
C :0x004D	00	NOP
C :0x004E	00	NOP
C :0x004F	00	NOP
C :0x0050	00	NOP
C :0x0051	00	NOP
C :0x0052	00	NOP
C :0x0053	00	NOP
C :0x0054	00	NOP
C :0x0055	00	NOP
C :0x0056	00	NOP
C :0x0057	00	NOP
C :0x0058	00	NOP
C :0x0059	00	NOP
C :0x005A	00	NOP
C :0x005B	00	NOP
C :0x005C	00	NOP
C :0x005D	00	NOP
C :0x005E	00	NOP
C :0x005F	00	NOP
C :0x0060	00	NOP
C :0x0061	00	NOP
C :0x0062	00	NOP
C :0x0063	00	NOP
C :0x0064	00	NOP
C :0x0065	00	NOP
C :0x0066	00	NOP
C :0x0067	00	NOP
C :0x0068	00	NOP
C :0x0069	00	NOP
C :0x006A	00	NOP
C :0x006B	00	NOP
C :0x006C	00	NOP
C :0x006D	00	NOP
C :0x006E	00	NOP
C :0x006F	00	NOP
C :0x0070	00	NOP
C :0x0071	00	NOP
C :0x0072	00	NOP
C :0x0073	00	NOP
C :0x0074	00	NOP
C :0x0075	00	NOP
C :0x0076	00	NOP
C :0x0077	00	NOP
C :0x0078	00	NOP
C :0x0079	00	NOP
C :0x007A	00	NOP
C :0x007B	00	NOP
C :0x007C	00	NOP
C :0x007D	00	NOP
C :0x007E	00	NOP
C :0x007F	00	NOP
C :0x0080	00	NOP
C :0x0081	00	NOP
C :0x0082	00	NOP
C :0x0083	00	NOP
C :0x0084	00	NOP
C :0x0085	00	NOP
C :0x0086	00	NOP
C :0x0087	00	NOP
C :0x0088	00	NOP
C :0x0089	00	NOP
C :0x008A	00	NOP
C :0x008B	00	NOP
C :0x008C	00	NOP
C :0x008D	00	NOP
C :0x008E	00	NOP

C :0x008F	00	NOP
C :0x0090	00	NOP
C :0x0091	00	NOP
C :0x0092	00	NOP
C :0x0093	00	NOP
C :0x0094	00	NOP
C :0x0095	00	NOP
C :0x0096	00	NOP
C :0x0097	00	NOP
C :0x0098	00	NOP
C :0x0099	00	NOP
C :0x009A	00	NOP
C :0x009B	00	NOP
C :0x009C	00	NOP
C :0x009D	00	NOP
C :0x009E	00	NOP
C :0x009F	00	NOP
C :0x00A0	00	NOP
C :0x00A1	00	NOP
C :0x00A2	00	NOP
C :0x00A3	00	NOP
C :0x00A4	00	NOP
C :0x00A5	00	NOP
C :0x00A6	00	NOP
C :0x00A7	00	NOP
C :0x00A8	00	NOP
C :0x00A9	00	NOP
C :0x00AA	00	NOP
C :0x00AB	00	NOP
C :0x00AC	00	NOP
C :0x00AD	00	NOP
C :0x00AE	00	NOP
C :0x00AF	00	NOP
C :0x00B0	00	NOP
C :0x00B1	00	NOP
C :0x00B2	00	NOP
C :0x00B3	00	NOP
C :0x00B4	00	NOP
C :0x00B5	00	NOP
C :0x00B6	00	NOP
C :0x00B7	00	NOP
C :0x00B8	00	NOP
C :0x00B9	00	NOP
C :0x00BA	00	NOP
C :0x00BB	00	NOP
C :0x00BC	00	NOP
C :0x00BD	00	NOP
C :0x00BE	00	NOP
C :0x00BF	00	NOP
C :0x00C0	00	NOP
C :0x00C1	00	NOP
C :0x00C2	00	NOP
C :0x00C3	00	NOP
C :0x00C4	00	NOP
C :0x00C5	00	NOP
C :0x00C6	00	NOP
C :0x00C7	00	NOP
C :0x00C8	00	NOP
C :0x00C9	00	NOP
C :0x00CA	00	NOP
C :0x00CB	00	NOP
C :0x00CC	00	NOP
C :0x00CD	00	NOP
C :0x00CE	00	NOP
C :0x00CF	00	NOP
C :0x00DO	00	NOP
C :0x00D1	00	NOP
C :0x00D2	00	NOP
C :0x00D3	00	NOP
C :0x00D4	00	NOP
C :0x00D5	00	NOP
C :0x00D6	00	NOP
C :0x00D7	00	NOP
C :0x00D8	00	NOP
C :0x00D9	00	NOP
C :0x00DA	00	NOP
C :0x00DB	00	NOP
C :0x00DC	00	NOP
C :0x00DD	00	NOP
C :0x00DE	00	NOP
C :0x00DF	00	NOP
C :0x00E0	00	NOP
C :0x00E1	00	NOP
C :0x00E2	00	NOP
C :0x00E3	00	NOP
C :0x00E4	00	NOP
C :0x00E5	00	NOP
C :0x00E6	00	NOP
C :0x00E7	00	NOP
C :0x00E8	00	NOP
C :0x00E9	00	NOP
C :0x00EA	00	NOP
C :0x00EB	00	NOP
C :0x00EC	00	NOP
C :0x00ED	00	NOP
C :0x00EE	00	NOP
C :0x00EF	00	NOP
C :0x00F0	00	NOP
C :0x00F1	00	NOP

C:0x00F2	00	NOP
C:0x00F3	00	NOP
C:0x00F4	00	NOP
C:0x00F5	00	NOP
C:0x00F6	00	NOP
C:0x00F7	00	NOP
C:0x00F8	00	NOP
C:0x00F9	00	NOP
C:0x00FA	00	NOP
C:0x00FB	00	NOP
C:0x00FC	00	NOP
C:0x00FD	00	NOP
C:0x00FE	00	NOP
C:0x00FF	00	NOP
C:0x0100	00	NOP
C:0x0101	00	NOP
C:0x0102	00	NOP
C:0x0103	00	NOP
C:0x0104	00	NOP
C:0x0105	00	NOP
C:0x0106	00	NOP
C:0x0107	00	NOP
C:0x0108	00	NOP
C:0x0109	00	NOP
C:0x010A	00	NOP
C:0x010B	00	NOP
C:0x010C	00	NOP
C:0x010D	00	NOP
C:0x010E	00	NOP
C:0x010F	00	NOP
C:0x0110	00	NOP
C:0x0111	00	NOP
C:0x0112	00	NOP
C:0x0113	00	NOP
C:0x0114	00	NOP
C:0x0115	00	NOP
C:0x0116	00	NOP
C:0x0117	00	NOP
C:0x0118	00	NOP
C:0x0119	00	NOP
C:0x011A	00	NOP
C:0x011B	00	NOP
C:0x011C	00	NOP
C:0x011D	00	NOP
C:0x011E	00	NOP
C:0x011F	00	NOP
C:0x0120	00	NOP
C:0x0121	00	NOP
C:0x0122	00	NOP
C:0x0123	00	NOP
C:0x0124	00	NOP
C:0x0125	00	NOP
C:0x0126	00	NOP
C:0x0127	00	NOP
C:0x0128	00	NOP
C:0x0129	00	NOP
C:0x012A	00	NOP
C:0x012B	00	NOP
C:0x012C	00	NOP
C:0x012D	00	NOP
C:0x012E	00	NOP
C:0x012F	00	NOP
C:0x0130	00	NOP
C:0x0131	00	NOP
C:0x0132	00	NOP
C:0x0133	00	NOP
C:0x0134	00	NOP
C:0x0135	00	NOP
C:0x0136	00	NOP
C:0x0137	00	NOP
C:0x0138	00	NOP
C:0x0139	00	NOP
C:0x013A	00	NOP
C:0x013B	00	NOP
C:0x013C	00	NOP
C:0x013D	00	NOP
C:0x013E	00	NOP
C:0x013F	00	NOP
C:0x0140	00	NOP
C:0x0141	00	NOP
C:0x0142	00	NOP
C:0x0143	00	NOP
C:0x0144	00	NOP
C:0x0145	00	NOP
C:0x0146	00	NOP
C:0x0147	00	NOP
C:0x0148	00	NOP
C:0x0149	00	NOP
C:0x014A	00	NOP
C:0x014B	00	NOP
C:0x014C	00	NOP
C:0x014D	00	NOP
C:0x014E	00	NOP
C:0x014F	00	NOP
C:0x0150	00	NOP
C:0x0151	00	NOP
C:0x0152	00	NOP
C:0x0153	00	NOP
C:0x0154	00	NOP

C:0x0155	00	NOP
C:0x0156	00	NOP
C:0x0157	00	NOP
C:0x0158	00	NOP
C:0x0159	00	NOP
C:0x015A	00	NOP
C:0x015B	00	NOP
C:0x015C	00	NOP
C:0x015D	00	NOP
C:0x015E	00	NOP
C:0x015F	00	NOP
C:0x0160	00	NOP
C:0x0161	00	NOP
C:0x0162	00	NOP
C:0x0163	00	NOP
C:0x0164	00	NOP
C:0x0165	00	NOP
C:0x0166	00	NOP
C:0x0167	00	NOP
C:0x0168	00	NOP
C:0x0169	00	NOP
C:0x016A	00	NOP
C:0x016B	00	NOP
C:0x016C	00	NOP
C:0x016D	00	NOP
C:0x016E	00	NOP
C:0x016F	00	NOP
C:0x0170	00	NOP
C:0x0171	00	NOP
C:0x0172	00	NOP
C:0x0173	00	NOP
C:0x0174	00	NOP
C:0x0175	00	NOP
C:0x0176	00	NOP
C:0x0177	00	NOP
C:0x0178	00	NOP
C:0x0179	00	NOP
C:0x017A	00	NOP
C:0x017B	00	NOP
C:0x017C	00	NOP
C:0x017D	00	NOP
C:0x017E	00	NOP
C:0x017F	00	NOP
C:0x0180	00	NOP
C:0x0181	00	NOP
C:0x0182	00	NOP
C:0x0183	00	NOP
C:0x0184	00	NOP
C:0x0185	00	NOP
C:0x0186	00	NOP
C:0x0187	00	NOP
C:0x0188	00	NOP
C:0x0189	00	NOP
C:0x018A	00	NOP
C:0x018B	00	NOP
C:0x018C	00	NOP
C:0x018D	00	NOP
C:0x018E	00	NOP
C:0x018F	00	NOP
C:0x0190	00	NOP
C:0x0191	00	NOP
C:0x0192	00	NOP
C:0x0193	00	NOP
C:0x0194	00	NOP
C:0x0195	00	NOP
C:0x0196	00	NOP
C:0x0197	00	NOP
C:0x0198	00	NOP
C:0x0199	00	NOP
C:0x019A	00	NOP
C:0x019B	00	NOP
C:0x019C	00	NOP
C:0x019D	00	NOP
C:0x019E	00	NOP
C:0x019F	00	NOP
C:0x01A0	00	NOP
C:0x01A1	00	NOP
C:0x01A2	00	NOP
C:0x01A3	00	NOP
C:0x01A4	00	NOP
C:0x01A5	00	NOP
C:0x01A6	00	NOP
C:0x01A7	00	NOP
C:0x01A8	00	NOP
C:0x01A9	00	NOP
C:0x01AA	00	NOP
C:0x01AB	00	NOP
C:0x01AC	00	NOP
C:0x01AD	00	NOP
C:0x01AE	00	NOP
C:0x01AF	00	NOP
C:0x01B0	00	NOP
C:0x01B1	00	NOP
C:0x01B2	00	NOP
C:0x01B3	00	NOP
C:0x01B4	00	NOP
C:0x01B5	00	NOP
C:0x01B6	00	NOP
C:0x01B7	00	NOP

C :0x01B8	00	NOP
C :0x01B9	00	NOP
C :0x01BA	00	NOP
C :0x01BB	00	NOP
C :0x01BC	00	NOP
C :0x01BD	00	NOP
C :0x01BE	00	NOP
C :0x01BF	00	NOP
C :0x01C0	00	NOP
C :0x01C1	00	NOP
C :0x01C2	00	NOP
C :0x01C3	00	NOP
C :0x01C4	00	NOP
C :0x01C5	00	NOP
C :0x01C6	00	NOP
C :0x01C7	00	NOP
C :0x01C8	00	NOP
C :0x01C9	00	NOP
C :0x01CA	00	NOP
C :0x01CB	00	NOP
C :0x01CC	00	NOP
C :0x01CD	00	NOP
C :0x01CE	00	NOP
C :0x01CF	00	NOP
C :0x01D0	00	NOP
C :0x01D1	00	NOP
C :0x01D2	00	NOP
C :0x01D3	00	NOP
C :0x01D4	00	NOP
C :0x01D5	00	NOP
C :0x01D6	00	NOP
C :0x01D7	00	NOP
C :0x01D8	00	NOP
C :0x01D9	00	NOP
C :0x01DA	00	NOP
C :0x01DB	00	NOP
C :0x01DC	00	NOP
C :0x01DD	00	NOP
C :0x01DE	00	NOP
C :0x01DF	00	NOP
C :0x01E0	00	NOP
C :0x01E1	00	NOP
C :0x01E2	00	NOP
C :0x01E3	00	NOP
C :0x01E4	00	NOP
C :0x01E5	00	NOP
C :0x01E6	00	NOP
C :0x01E7	00	NOP
C :0x01E8	00	NOP
C :0x01E9	00	NOP
C :0x01EA	00	NOP
C :0x01EB	00	NOP
C :0x01EC	00	NOP
C :0x01ED	00	NOP
C :0x01EE	00	NOP
C :0x01EF	00	NOP
C :0x01F0	00	NOP
C :0x01F1	00	NOP
C :0x01F2	00	NOP
C :0x01F3	00	NOP
C :0x01F4	00	NOP
C :0x01F5	00	NOP
C :0x01F6	00	NOP
C :0x01F7	00	NOP
C :0x01F8	00	NOP
C :0x01F9	00	NOP
C :0x01FA	00	NOP
C :0x01FB	00	NOP
C :0x01FC	00	NOP
C :0x01FD	00	NOP
C :0x01FE	00	NOP
C :0x01FF	00	NOP
C :0x0200	00	NOP
C :0x0201	00	NOP
C :0x0202	00	NOP
C :0x0203	00	NOP
C :0x0204	00	NOP
C :0x0205	00	NOP
C :0x0206	00	NOP
C :0x0207	00	NOP
C :0x0208	00	NOP
C :0x0209	00	NOP
C :0x020A	00	NOP
C :0x020B	00	NOP
C :0x020C	00	NOP
C :0x020D	00	NOP
C :0x020E	00	NOP
C :0x020F	00	NOP
C :0x0210	00	NOP
C :0x0211	00	NOP
C :0x0212	00	NOP
C :0x0213	00	NOP
C :0x0214	00	NOP
C :0x0215	00	NOP
C :0x0216	00	NOP
C :0x0217	00	NOP
C :0x0218	00	NOP
C :0x0219	00	NOP
C :0x021A	00	NOP

C :0x021B	00	NOP
C :0x021C	00	NOP
C :0x021D	00	NOP
C :0x021E	00	NOP
C :0x021F	00	NOP
C :0x0220	00	NOP
C :0x0221	00	NOP
C :0x0222	00	NOP
C :0x0223	00	NOP
C :0x0224	00	NOP
C :0x0225	00	NOP
C :0x0226	00	NOP
C :0x0227	00	NOP
C :0x0228	00	NOP
C :0x0229	00	NOP
C :0x022A	00	NOP
C :0x022B	00	NOP
C :0x022C	00	NOP
C :0x022D	00	NOP
C :0x022E	00	NOP
C :0x022F	00	NOP
C :0x0230	00	NOP
C :0x0231	00	NOP
C :0x0232	00	NOP
C :0x0233	00	NOP
C :0x0234	00	NOP
C :0x0235	00	NOP
C :0x0236	00	NOP
C :0x0237	00	NOP
C :0x0238	00	NOP
C :0x0239	00	NOP
C :0x023A	00	NOP
C :0x023B	00	NOP
C :0x023C	00	NOP
C :0x023D	00	NOP
C :0x023E	00	NOP
C :0x023F	00	NOP
C :0x0240	00	NOP
C :0x0241	00	NOP
C :0x0242	00	NOP
C :0x0243	00	NOP
C :0x0244	00	NOP
C :0x0245	00	NOP
C :0x0246	00	NOP
C :0x0247	00	NOP
C :0x0248	00	NOP
C :0x0249	00	NOP
C :0x024A	00	NOP
C :0x024B	00	NOP
C :0x024C	00	NOP
C :0x024D	00	NOP
C :0x024E	00	NOP
C :0x024F	00	NOP
C :0x0250	00	NOP
C :0x0251	00	NOP
C :0x0252	00	NOP
C :0x0253	00	NOP
C :0x0254	00	NOP
C :0x0255	00	NOP
C :0x0256	00	NOP
C :0x0257	00	NOP
C :0x0258	00	NOP
C :0x0259	00	NOP
C :0x025A	00	NOP
C :0x025B	00	NOP
C :0x025C	00	NOP
C :0x025D	00	NOP
C :0x025E	00	NOP
C :0x025F	00	NOP
C :0x0260	00	NOP
C :0x0261	00	NOP
C :0x0262	00	NOP
C :0x0263	00	NOP
C :0x0264	00	NOP
C :0x0265	00	NOP
C :0x0266	00	NOP
C :0x0267	00	NOP
C :0x0268	00	NOP
C :0x0269	00	NOP
C :0x026A	00	NOP
C :0x026B	00	NOP
C :0x026C	00	NOP
C :0x026D	00	NOP
C :0x026E	00	NOP
C :0x026F	00	NOP
C :0x0270	00	NOP
C :0x0271	00	NOP
C :0x0272	00	NOP
C :0x0273	00	NOP
C :0x0274	00	NOP
C :0x0275	00	NOP
C :0x0276	00	NOP
C :0x0277	00	NOP
C :0x0278	00	NOP
C :0x0279	00	NOP
C :0x027A	00	NOP
C :0x027B	00	NOP
C :0x027C	00	NOP
C :0x027D	00	NOP

C :0x027E	00	NOP
C :0x027F	00	NOP
C :0x0280	00	NOP
C :0x0281	00	NOP
C :0x0282	00	NOP
C :0x0283	00	NOP
C :0x0284	00	NOP
C :0x0285	00	NOP
C :0x0286	00	NOP
C :0x0287	00	NOP
C :0x0288	00	NOP
C :0x0289	00	NOP
C :0x028A	00	NOP
C :0x028B	00	NOP
C :0x028C	00	NOP
C :0x028D	00	NOP
C :0x028E	00	NOP
C :0x028F	00	NOP
C :0x0290	00	NOP
C :0x0291	00	NOP
C :0x0292	00	NOP
C :0x0293	00	NOP
C :0x0294	00	NOP
C :0x0295	00	NOP
C :0x0296	00	NOP
C :0x0297	00	NOP
C :0x0298	00	NOP
C :0x0299	00	NOP
C :0x029A	00	NOP
C :0x029B	00	NOP
C :0x029C	00	NOP
C :0x029D	00	NOP
C :0x029E	00	NOP
C :0x029F	00	NOP
C :0x02A0	00	NOP
C :0x02A1	00	NOP
C :0x02A2	00	NOP
C :0x02A3	00	NOP
C :0x02A4	00	NOP
C :0x02A5	00	NOP
C :0x02A6	00	NOP
C :0x02A7	00	NOP
C :0x02A8	00	NOP
C :0x02A9	00	NOP
C :0x02AA	00	NOP
C :0x02AB	00	NOP
C :0x02AC	00	NOP
C :0x02AD	00	NOP
C :0x02AE	00	NOP
C :0x02AF	00	NOP
C :0x02B0	00	NOP
C :0x02B1	00	NOP
C :0x02B2	00	NOP
C :0x02B3	00	NOP
C :0x02B4	00	NOP
C :0x02B5	00	NOP
C :0x02B6	00	NOP
C :0x02B7	00	NOP
C :0x02B8	00	NOP
C :0x02B9	00	NOP
C :0x02BA	00	NOP
C :0x02BB	00	NOP
C :0x02BC	00	NOP
C :0x02BD	00	NOP
C :0x02BE	00	NOP
C :0x02BF	00	NOP
C :0x02C0	00	NOP
C :0x02C1	00	NOP
C :0x02C2	00	NOP
C :0x02C3	00	NOP
C :0x02C4	00	NOP
C :0x02C5	00	NOP
C :0x02C6	00	NOP
C :0x02C7	00	NOP
C :0x02C8	00	NOP
C :0x02C9	00	NOP
C :0x02CA	00	NOP
C :0x02CB	00	NOP
C :0x02CC	00	NOP
C :0x02CD	00	NOP
C :0x02CE	00	NOP
C :0x02CF	00	NOP
C :0x02D0	00	NOP
C :0x02D1	00	NOP
C :0x02D2	00	NOP
C :0x02D3	00	NOP
C :0x02D4	00	NOP
C :0x02D5	00	NOP
C :0x02D6	00	NOP
C :0x02D7	00	NOP
C :0x02D8	00	NOP
C :0x02D9	00	NOP
C :0x02DA	00	NOP
C :0x02DB	00	NOP
C :0x02DC	00	NOP
C :0x02DD	00	NOP
C :0x02DE	00	NOP
C :0x02DF	00	NOP
C :0x02E0	00	NOP

C :0x02E1	00	NOP
C :0x02E2	00	NOP
C :0x02E3	00	NOP
C :0x02E4	00	NOP
C :0x02E5	00	NOP
C :0x02E6	00	NOP
C :0x02E7	00	NOP
C :0x02E8	00	NOP
C :0x02E9	00	NOP
C :0x02EA	00	NOP
C :0x02EB	00	NOP
C :0x02EC	00	NOP
C :0x02ED	00	NOP
C :0x02EE	00	NOP
C :0x02EF	00	NOP
C :0x02F0	00	NOP
C :0x02F1	00	NOP
C :0x02F2	00	NOP
C :0x02F3	00	NOP
C :0x02F4	00	NOP
C :0x02F5	00	NOP
C :0x02F6	00	NOP
C :0x02F7	00	NOP
C :0x02F8	00	NOP
C :0x02F9	00	NOP
C :0x02FA	00	NOP
C :0x02FB	00	NOP
C :0x02FC	00	NOP
C :0x02FD	00	NOP
C :0x02FE	00	NOP
C :0x02FF	00	NOP
C :0x0300	00	NOP
C :0x0301	00	NOP
C :0x0302	00	NOP
C :0x0303	00	NOP
C :0x0304	00	NOP
C :0x0305	00	NOP
C :0x0306	00	NOP
C :0x0307	00	NOP
C :0x0308	00	NOP
C :0x0309	00	NOP
C :0x030A	00	NOP
C :0x030B	00	NOP
C :0x030C	00	NOP
C :0x030D	00	NOP
C :0x030E	00	NOP
C :0x030F	00	NOP
C :0x0310	00	NOP
C :0x0311	00	NOP
C :0x0312	00	NOP
C :0x0313	00	NOP
C :0x0314	00	NOP
C :0x0315	00	NOP
C :0x0316	00	NOP
C :0x0317	00	NOP
C :0x0318	00	NOP
C :0x0319	00	NOP
C :0x031A	00	NOP
C :0x031B	00	NOP
C :0x031C	00	NOP
C :0x031D	00	NOP
C :0x031E	00	NOP
C :0x031F	00	NOP
C :0x0320	00	NOP
C :0x0321	00	NOP
C :0x0322	00	NOP
C :0x0323	00	NOP
C :0x0324	00	NOP
C :0x0325	00	NOP
C :0x0326	00	NOP
C :0x0327	00	NOP
C :0x0328	00	NOP
C :0x0329	00	NOP
C :0x032A	00	NOP
C :0x032B	00	NOP
C :0x032C	00	NOP
C :0x032D	00	NOP
C :0x032E	00	NOP
C :0x032F	00	NOP
C :0x0330	00	NOP
C :0x0331	00	NOP
C :0x0332	00	NOP
C :0x0333	00	NOP
C :0x0334	00	NOP
C :0x0335	00	NOP
C :0x0336	00	NOP
C :0x0337	00	NOP
C :0x0338	00	NOP
C :0x0339	00	NOP
C :0x033A	00	NOP
C :0x033B	00	NOP
C :0x033C	00	NOP
C :0x033D	00	NOP
C :0x033E	00	NOP
C :0x033F	00	NOP
C :0x0340	00	NOP
C :0x0341	00	NOP
C :0x0342	00	NOP
C :0x0343	00	NOP

C :0x0344	00	NOP
C :0x0345	00	NOP
C :0x0346	00	NOP
C :0x0347	00	NOP
C :0x0348	00	NOP
C :0x0349	00	NOP
C :0x034A	00	NOP
C :0x034B	00	NOP
C :0x034C	00	NOP
C :0x034D	00	NOP
C :0x034E	00	NOP
C :0x034F	00	NOP
C :0x0350	00	NOP
C :0x0351	00	NOP
C :0x0352	00	NOP
C :0x0353	00	NOP
C :0x0354	00	NOP
C :0x0355	00	NOP
C :0x0356	00	NOP
C :0x0357	00	NOP
C :0x0358	00	NOP
C :0x0359	00	NOP
C :0x035A	00	NOP
C :0x035B	00	NOP
C :0x035C	00	NOP
C :0x035D	00	NOP
C :0x035E	00	NOP
C :0x035F	00	NOP
C :0x0360	00	NOP
C :0x0361	00	NOP
C :0x0362	00	NOP
C :0x0363	00	NOP
C :0x0364	00	NOP
C :0x0365	00	NOP
C :0x0366	00	NOP
C :0x0367	00	NOP
C :0x0368	00	NOP
C :0x0369	00	NOP
C :0x036A	00	NOP
C :0x036B	00	NOP
C :0x036C	00	NOP
C :0x036D	00	NOP
C :0x036E	00	NOP
C :0x036F	00	NOP
C :0x0370	00	NOP
C :0x0371	00	NOP
C :0x0372	00	NOP
C :0x0373	00	NOP
C :0x0374	00	NOP
C :0x0375	00	NOP
C :0x0376	00	NOP
C :0x0377	00	NOP
C :0x0378	00	NOP
C :0x0379	00	NOP
C :0x037A	00	NOP
C :0x037B	00	NOP
C :0x037C	00	NOP
C :0x037D	00	NOP
C :0x037E	00	NOP
C :0x037F	00	NOP
C :0x0380	00	NOP
C :0x0381	00	NOP
C :0x0382	00	NOP
C :0x0383	00	NOP
C :0x0384	00	NOP
C :0x0385	00	NOP
C :0x0386	00	NOP
C :0x0387	00	NOP
C :0x0388	00	NOP
C :0x0389	00	NOP
C :0x038A	00	NOP
C :0x038B	00	NOP
C :0x038C	00	NOP
C :0x038D	00	NOP
C :0x038E	00	NOP
C :0x038F	00	NOP
C :0x0390	00	NOP
C :0x0391	00	NOP
C :0x0392	00	NOP
C :0x0393	00	NOP
C :0x0394	00	NOP
C :0x0395	00	NOP
C :0x0396	00	NOP
C :0x0397	00	NOP
C :0x0398	00	NOP
C :0x0399	00	NOP
C :0x039A	00	NOP
C :0x039B	00	NOP
C :0x039C	00	NOP
C :0x039D	00	NOP
C :0x039E	00	NOP
C :0x039F	00	NOP
C :0x03A0	00	NOP
C :0x03A1	00	NOP
C :0x03A2	00	NOP
C :0x03A3	00	NOP
C :0x03A4	00	NOP
C :0x03A5	00	NOP
C :0x03A6	00	NOP

C :0x03A7	00	NOP
C :0x03A8	00	NOP
C :0x03A9	00	NOP
C :0x03AA	00	NOP
C :0x03AB	00	NOP
C :0x03AC	00	NOP
C :0x03AD	00	NOP
C :0x03AE	00	NOP
C :0x03AF	00	NOP
C :0x03B0	00	NOP
C :0x03B1	00	NOP
C :0x03B2	00	NOP
C :0x03B3	00	NOP
C :0x03B4	00	NOP
C :0x03B5	00	NOP
C :0x03B6	00	NOP
C :0x03B7	00	NOP
C :0x03B8	00	NOP
C :0x03B9	00	NOP
C :0x03BA	00	NOP
C :0x03BB	00	NOP
C :0x03BC	00	NOP
C :0x03BD	00	NOP
C :0x03BE	00	NOP
C :0x03BF	00	NOP
C :0x03C0	00	NOP
C :0x03C1	00	NOP
C :0x03C2	00	NOP
C :0x03C3	00	NOP
C :0x03C4	00	NOP
C :0x03C5	00	NOP
C :0x03C6	00	NOP
C :0x03C7	00	NOP
C :0x03C8	00	NOP
C :0x03C9	00	NOP
C :0x03CA	00	NOP
C :0x03CB	00	NOP
C :0x03CC	00	NOP
C :0x03CD	00	NOP
C :0x03CE	00	NOP
C :0x03CF	00	NOP
C :0x03D0	00	NOP
C :0x03D1	00	NOP
C :0x03D2	00	NOP
C :0x03D3	00	NOP
C :0x03D4	00	NOP
C :0x03D5	00	NOP
C :0x03D6	00	NOP
C :0x03D7	00	NOP
C :0x03D8	00	NOP
C :0x03D9	00	NOP
C :0x03DA	00	NOP
C :0x03DB	00	NOP
C :0x03DC	00	NOP
C :0x03DD	00	NOP
C :0x03DE	00	NOP
C :0x03DF	00	NOP
C :0x03E0	00	NOP
C :0x03E1	00	NOP
C :0x03E2	00	NOP
C :0x03E3	00	NOP
C :0x03E4	00	NOP
C :0x03E5	00	NOP
C :0x03E6	00	NOP
C :0x03E7	00	NOP
C :0x03E8	00	NOP
C :0x03E9	00	NOP
C :0x03EA	00	NOP
C :0x03EB	00	NOP
C :0x03EC	00	NOP
C :0x03ED	00	NOP
C :0x03EE	00	NOP
C :0x03EF	00	NOP
C :0x03F0	00	NOP
C :0x03F1	00	NOP
C :0x03F2	00	NOP
C :0x03F3	00	NOP
C :0x03F4	00	NOP
C :0x03F5	00	NOP
C :0x03F6	00	NOP
C :0x03F7	00	NOP
C :0x03F8	00	NOP
C :0x03F9	00	NOP
C :0x03FA	00	NOP
C :0x03FB	00	NOP
C :0x03FC	00	NOP
C :0x03FD	00	NOP
C :0x03FE	00	NOP
C :0x03FF	00	NOP
C :0x0400	00	NOP
C :0x0401	00	NOP
C :0x0402	00	NOP
C :0x0403	00	NOP
C :0x0404	00	NOP
C :0x0405	00	NOP
C :0x0406	00	NOP
C :0x0407	00	NOP
C :0x0408	00	NOP
C :0x0409	00	NOP

C :0x040A	00	NOP
C :0x040B	00	NOP
C :0x040C	00	NOP
C :0x040D	00	NOP
C :0x040E	00	NOP
C :0x040F	00	NOP
C :0x0410	00	NOP
C :0x0411	00	NOP
C :0x0412	00	NOP
C :0x0413	00	NOP
C :0x0414	00	NOP
C :0x0415	00	NOP
C :0x0416	00	NOP
C :0x0417	00	NOP
C :0x0418	00	NOP
C :0x0419	00	NOP
C :0x041A	00	NOP
C :0x041B	00	NOP
C :0x041C	00	NOP
C :0x041D	00	NOP
C :0x041E	00	NOP
C :0x041F	00	NOP
C :0x0420	00	NOP
C :0x0421	00	NOP
C :0x0422	00	NOP
C :0x0423	00	NOP
C :0x0424	00	NOP
C :0x0425	00	NOP
C :0x0426	00	NOP
C :0x0427	00	NOP
C :0x0428	00	NOP
C :0x0429	00	NOP
C :0x042A	00	NOP
C :0x042B	00	NOP
C :0x042C	00	NOP
C :0x042D	00	NOP
C :0x042E	00	NOP
C :0x042F	00	NOP
C :0x0430	00	NOP
C :0x0431	00	NOP
C :0x0432	00	NOP
C :0x0433	00	NOP
C :0x0434	00	NOP
C :0x0435	00	NOP
C :0x0436	00	NOP
C :0x0437	00	NOP
C :0x0438	00	NOP
C :0x0439	00	NOP
C :0x043A	00	NOP
C :0x043B	00	NOP
C :0x043C	00	NOP
C :0x043D	00	NOP
C :0x043E	00	NOP
C :0x043F	00	NOP
C :0x0440	00	NOP
C :0x0441	00	NOP
C :0x0442	00	NOP
C :0x0443	00	NOP
C :0x0444	00	NOP
C :0x0445	00	NOP
C :0x0446	00	NOP
C :0x0447	00	NOP
C :0x0448	00	NOP
C :0x0449	00	NOP
C :0x044A	00	NOP
C :0x044B	00	NOP
C :0x044C	00	NOP
C :0x044D	00	NOP
C :0x044E	00	NOP
C :0x044F	00	NOP
C :0x0450	00	NOP
C :0x0451	00	NOP
C :0x0452	00	NOP
C :0x0453	00	NOP
C :0x0454	00	NOP
C :0x0455	00	NOP
C :0x0456	00	NOP
C :0x0457	00	NOP
C :0x0458	00	NOP
C :0x0459	00	NOP
C :0x045A	00	NOP
C :0x045B	00	NOP
C :0x045C	00	NOP
C :0x045D	00	NOP
C :0x045E	00	NOP
C :0x045F	00	NOP
C :0x0460	00	NOP
C :0x0461	00	NOP
C :0x0462	00	NOP
C :0x0463	00	NOP
C :0x0464	00	NOP
C :0x0465	00	NOP
C :0x0466	00	NOP
C :0x0467	00	NOP
C :0x0468	00	NOP
C :0x0469	00	NOP
C :0x046A	00	NOP
C :0x046B	00	NOP
C :0x046C	00	NOP

C :0x046D	00	NOP
C :0x046E	00	NOP
C :0x046F	00	NOP
C :0x0470	00	NOP
C :0x0471	00	NOP
C :0x0472	00	NOP
C :0x0473	00	NOP
C :0x0474	00	NOP
C :0x0475	00	NOP
C :0x0476	00	NOP
C :0x0477	00	NOP
C :0x0478	00	NOP
C :0x0479	00	NOP
C :0x047A	00	NOP
C :0x047B	00	NOP
C :0x047C	00	NOP
C :0x047D	00	NOP
C :0x047E	00	NOP
C :0x047F	00	NOP
C :0x0480	00	NOP
C :0x0481	00	NOP
C :0x0482	00	NOP
C :0x0483	00	NOP
C :0x0484	00	NOP
C :0x0485	00	NOP
C :0x0486	00	NOP
C :0x0487	00	NOP
C :0x0488	00	NOP
C :0x0489	00	NOP
C :0x048A	00	NOP
C :0x048B	00	NOP
C :0x048C	00	NOP
C :0x048D	00	NOP
C :0x048E	00	NOP
C :0x048F	00	NOP
C :0x0490	00	NOP
C :0x0491	00	NOP
C :0x0492	00	NOP
C :0x0493	00	NOP
C :0x0494	00	NOP
C :0x0495	00	NOP
C :0x0496	00	NOP
C :0x0497	00	NOP
C :0x0498	00	NOP
C :0x0499	00	NOP
C :0x049A	00	NOP
C :0x049B	00	NOP
C :0x049C	00	NOP
C :0x049D	00	NOP
C :0x049E	00	NOP
C :0x049F	00	NOP
C :0x04A0	00	NOP
C :0x04A1	00	NOP
C :0x04A2	00	NOP
C :0x04A3	00	NOP
C :0x04A4	00	NOP
C :0x04A5	00	NOP
C :0x04A6	00	NOP
C :0x04A7	00	NOP
C :0x04A8	00	NOP
C :0x04A9	00	NOP
C :0x04AA	00	NOP
C :0x04AB	00	NOP
C :0x04AC	00	NOP
C :0x04AD	00	NOP
C :0x04AE	00	NOP
C :0x04AF	00	NOP
C :0x04B0	00	NOP
C :0x04B1	00	NOP
C :0x04B2	00	NOP
C :0x04B3	00	NOP
C :0x04B4	00	NOP
C :0x04B5	00	NOP
C :0x04B6	00	NOP
C :0x04B7	00	NOP
C :0x04B8	00	NOP
C :0x04B9	00	NOP
C :0x04BA	00	NOP
C :0x04BB	00	NOP
C :0x04BC	00	NOP
C :0x04BD	00	NOP
C :0x04BE	00	NOP
C :0x04BF	00	NOP
C :0x04C0	00	NOP
C :0x04C1	00	NOP
C :0x04C2	00	NOP
C :0x04C3	00	NOP
C :0x04C4	00	NOP
C :0x04C5	00	NOP
C :0x04C6	00	NOP
C :0x04C7	00	NOP
C :0x04C8	00	NOP
C :0x04C9	00	NOP
C :0x04CA	00	NOP
C :0x04CB	00	NOP
C :0x04CC	00	NOP
C :0x04CD	00	NOP
C :0x04CE	00	NOP
C :0x04CF	00	NOP

C :0x04D0	00	NOP
C :0x04D1	00	NOP
C :0x04D2	00	NOP
C :0x04D3	00	NOP
C :0x04D4	00	NOP
C :0x04D5	00	NOP
C :0x04D6	00	NOP
C :0x04D7	00	NOP
C :0x04D8	00	NOP
C :0x04D9	00	NOP
C :0x04DA	00	NOP
C :0x04DB	00	NOP
C :0x04DC	00	NOP
C :0x04DD	00	NOP
C :0x04DE	00	NOP
C :0x04DF	00	NOP
C :0x04E0	00	NOP
C :0x04E1	00	NOP
C :0x04E2	00	NOP
C :0x04E3	00	NOP
C :0x04E4	00	NOP
C :0x04E5	00	NOP
C :0x04E6	00	NOP
C :0x04E7	00	NOP
C :0x04E8	00	NOP
C :0x04E9	00	NOP
C :0x04EA	00	NOP
C :0x04EB	00	NOP
C :0x04EC	00	NOP
C :0x04ED	00	NOP
C :0x04EE	00	NOP
C :0x04EF	00	NOP
C :0x04F0	00	NOP
C :0x04F1	00	NOP
C :0x04F2	00	NOP
C :0x04F3	00	NOP
C :0x04F4	00	NOP
C :0x04F5	00	NOP
C :0x04F6	00	NOP
C :0x04F7	00	NOP
C :0x04F8	00	NOP
C :0x04F9	00	NOP
C :0x04FA	00	NOP
C :0x04FB	00	NOP
C :0x04FC	00	NOP
C :0x04FD	00	NOP
C :0x04FE	00	NOP
C :0x04FF	00	NOP
C :0x0500	00	NOP
C :0x0501	00	NOP
C :0x0502	00	NOP
C :0x0503	00	NOP
C :0x0504	00	NOP
C :0x0505	00	NOP
C :0x0506	00	NOP
C :0x0507	00	NOP
C :0x0508	00	NOP
C :0x0509	00	NOP
C :0x050A	00	NOP
C :0x050B	00	NOP
C :0x050C	00	NOP
C :0x050D	00	NOP
C :0x050E	00	NOP
C :0x050F	00	NOP
C :0x0510	00	NOP
C :0x0511	00	NOP
C :0x0512	00	NOP
C :0x0513	00	NOP
C :0x0514	00	NOP
C :0x0515	00	NOP
C :0x0516	00	NOP
C :0x0517	00	NOP
C :0x0518	00	NOP
C :0x0519	00	NOP
C :0x051A	00	NOP
C :0x051B	00	NOP
C :0x051C	00	NOP
C :0x051D	00	NOP
C :0x051E	00	NOP
C :0x051F	00	NOP
C :0x0520	00	NOP
C :0x0521	00	NOP
C :0x0522	00	NOP
C :0x0523	00	NOP
C :0x0524	00	NOP
C :0x0525	00	NOP
C :0x0526	00	NOP
C :0x0527	00	NOP
C :0x0528	00	NOP
C :0x0529	00	NOP
C :0x052A	00	NOP
C :0x052B	00	NOP
C :0x052C	00	NOP
C :0x052D	00	NOP
C :0x052E	00	NOP
C :0x052F	00	NOP
C :0x0530	00	NOP
C :0x0531	00	NOP
C :0x0532	00	NOP

C :0x0533	00	NOP
C :0x0534	00	NOP
C :0x0535	00	NOP
C :0x0536	00	NOP
C :0x0537	00	NOP
C :0x0538	00	NOP
C :0x0539	00	NOP
C :0x053A	00	NOP
C :0x053B	00	NOP
C :0x053C	00	NOP
C :0x053D	00	NOP
C :0x053E	00	NOP
C :0x053F	00	NOP
C :0x0540	00	NOP
C :0x0541	00	NOP
C :0x0542	00	NOP
C :0x0543	00	NOP
C :0x0544	00	NOP
C :0x0545	00	NOP
C :0x0546	00	NOP
C :0x0547	00	NOP
C :0x0548	00	NOP
C :0x0549	00	NOP
C :0x054A	00	NOP
C :0x054B	00	NOP
C :0x054C	00	NOP
C :0x054D	00	NOP
C :0x054E	00	NOP
C :0x054F	00	NOP
C :0x0550	00	NOP
C :0x0551	00	NOP
C :0x0552	00	NOP
C :0x0553	00	NOP
C :0x0554	00	NOP
C :0x0555	00	NOP
C :0x0556	00	NOP
C :0x0557	00	NOP
C :0x0558	00	NOP
C :0x0559	00	NOP
C :0x055A	00	NOP
C :0x055B	00	NOP
C :0x055C	00	NOP
C :0x055D	00	NOP
C :0x055E	00	NOP
C :0x055F	00	NOP
C :0x0560	00	NOP
C :0x0561	00	NOP
C :0x0562	00	NOP
C :0x0563	00	NOP
C :0x0564	00	NOP
C :0x0565	00	NOP
C :0x0566	00	NOP
C :0x0567	00	NOP
C :0x0568	00	NOP
C :0x0569	00	NOP
C :0x056A	00	NOP
C :0x056B	00	NOP
C :0x056C	00	NOP
C :0x056D	00	NOP
C :0x056E	00	NOP
C :0x056F	00	NOP
C :0x0570	00	NOP
C :0x0571	00	NOP
C :0x0572	00	NOP
C :0x0573	00	NOP
C :0x0574	00	NOP
C :0x0575	00	NOP
C :0x0576	00	NOP
C :0x0577	00	NOP
C :0x0578	00	NOP
C :0x0579	00	NOP
C :0x057A	00	NOP
C :0x057B	00	NOP
C :0x057C	00	NOP
C :0x057D	00	NOP
C :0x057E	00	NOP
C :0x057F	00	NOP
C :0x0580	00	NOP
C :0x0581	00	NOP
C :0x0582	00	NOP
C :0x0583	00	NOP
C :0x0584	00	NOP
C :0x0585	00	NOP
C :0x0586	00	NOP
C :0x0587	00	NOP
C :0x0588	00	NOP
C :0x0589	00	NOP
C :0x058A	00	NOP
C :0x058B	00	NOP
C :0x058C	00	NOP
C :0x058D	00	NOP
C :0x058E	00	NOP
C :0x058F	00	NOP
C :0x0590	00	NOP
C :0x0591	00	NOP
C :0x0592	00	NOP
C :0x0593	00	NOP
C :0x0594	00	NOP
C :0x0595	00	NOP

C :0x0596	00	NOP
C :0x0597	00	NOP
C :0x0598	00	NOP
C :0x0599	00	NOP
C :0x059A	00	NOP
C :0x059B	00	NOP
C :0x059C	00	NOP
C :0x059D	00	NOP
C :0x059E	00	NOP
C :0x059F	00	NOP
C :0x05A0	00	NOP
C :0x05A1	00	NOP
C :0x05A2	00	NOP
C :0x05A3	00	NOP
C :0x05A4	00	NOP
C :0x05A5	00	NOP
C :0x05A6	00	NOP
C :0x05A7	00	NOP
C :0x05A8	00	NOP
C :0x05A9	00	NOP
C :0x05AA	00	NOP
C :0x05AB	00	NOP
C :0x05AC	00	NOP
C :0x05AD	00	NOP
C :0x05AE	00	NOP
C :0x05AF	00	NOP
C :0x05B0	00	NOP
C :0x05B1	00	NOP
C :0x05B2	00	NOP
C :0x05B3	00	NOP
C :0x05B4	00	NOP
C :0x05B5	00	NOP
C :0x05B6	00	NOP
C :0x05B7	00	NOP
C :0x05B8	00	NOP
C :0x05B9	00	NOP
C :0x05BA	00	NOP
C :0x05BB	00	NOP
C :0x05BC	00	NOP
C :0x05BD	00	NOP
C :0x05BE	00	NOP
C :0x05BF	00	NOP
C :0x05C0	00	NOP
C :0x05C1	00	NOP
C :0x05C2	00	NOP
C :0x05C3	00	NOP
C :0x05C4	00	NOP
C :0x05C5	00	NOP
C :0x05C6	00	NOP
C :0x05C7	00	NOP
C :0x05C8	00	NOP
C :0x05C9	00	NOP
C :0x05CA	00	NOP
C :0x05CB	00	NOP
C :0x05CC	00	NOP
C :0x05CD	00	NOP
C :0x05CE	00	NOP
C :0x05CF	00	NOP
C :0x05D0	00	NOP
C :0x05D1	00	NOP
C :0x05D2	00	NOP
C :0x05D3	00	NOP
C :0x05D4	00	NOP
C :0x05D5	00	NOP
C :0x05D6	00	NOP
C :0x05D7	00	NOP
C :0x05D8	00	NOP
C :0x05D9	00	NOP
C :0x05DA	00	NOP
C :0x05DB	00	NOP
C :0x05DC	00	NOP
C :0x05DD	00	NOP
C :0x05DE	00	NOP
C :0x05DF	00	NOP
C :0x05E0	00	NOP
C :0x05E1	00	NOP
C :0x05E2	00	NOP
C :0x05E3	00	NOP
C :0x05E4	00	NOP
C :0x05E5	00	NOP
C :0x05E6	00	NOP
C :0x05E7	00	NOP
C :0x05E8	00	NOP
C :0x05E9	00	NOP
C :0x05EA	00	NOP
C :0x05EB	00	NOP
C :0x05EC	00	NOP
C :0x05ED	00	NOP
C :0x05EE	00	NOP
C :0x05EF	00	NOP
C :0x05F0	00	NOP
C :0x05F1	00	NOP
C :0x05F2	00	NOP
C :0x05F3	00	NOP
C :0x05F4	00	NOP
C :0x05F5	00	NOP
C :0x05F6	00	NOP
C :0x05F7	00	NOP
C :0x05F8	00	NOP

C :0x05F9	00	NOP
C :0x05FA	00	NOP
C :0x05FB	00	NOP
C :0x05FC	00	NOP
C :0x05FD	00	NOP
C :0x05FE	00	NOP
C :0x05FF	00	NOP
C :0x0600	00	NOP
C :0x0601	00	NOP
C :0x0602	00	NOP
C :0x0603	00	NOP
C :0x0604	00	NOP
C :0x0605	00	NOP
C :0x0606	00	NOP
C :0x0607	00	NOP
C :0x0608	00	NOP
C :0x0609	00	NOP
C :0x060A	00	NOP
C :0x060B	00	NOP
C :0x060C	00	NOP
C :0x060D	00	NOP
C :0x060E	00	NOP
C :0x060F	00	NOP
C :0x0610	00	NOP
C :0x0611	00	NOP
C :0x0612	00	NOP
C :0x0613	00	NOP
C :0x0614	00	NOP
C :0x0615	00	NOP
C :0x0616	00	NOP
C :0x0617	00	NOP
C :0x0618	00	NOP
C :0x0619	00	NOP
C :0x061A	00	NOP
C :0x061B	00	NOP
C :0x061C	00	NOP
C :0x061D	00	NOP
C :0x061E	00	NOP
C :0x061F	00	NOP
C :0x0620	00	NOP
C :0x0621	00	NOP
C :0x0622	00	NOP
C :0x0623	00	NOP
C :0x0624	00	NOP
C :0x0625	00	NOP
C :0x0626	00	NOP
C :0x0627	00	NOP
C :0x0628	00	NOP
C :0x0629	00	NOP
C :0x062A	00	NOP
C :0x062B	00	NOP
C :0x062C	00	NOP
C :0x062D	00	NOP
C :0x062E	00	NOP
C :0x062F	00	NOP
C :0x0630	00	NOP
C :0x0631	00	NOP
C :0x0632	00	NOP
C :0x0633	00	NOP
C :0x0634	00	NOP
C :0x0635	00	NOP
C :0x0636	00	NOP
C :0x0637	00	NOP
C :0x0638	00	NOP
C :0x0639	00	NOP
C :0x063A	00	NOP
C :0x063B	00	NOP
C :0x063C	00	NOP
C :0x063D	00	NOP
C :0x063E	00	NOP
C :0x063F	00	NOP
C :0x0640	00	NOP
C :0x0641	00	NOP
C :0x0642	00	NOP
C :0x0643	00	NOP
C :0x0644	00	NOP
C :0x0645	00	NOP
C :0x0646	00	NOP
C :0x0647	00	NOP
C :0x0648	00	NOP
C :0x0649	00	NOP
C :0x064A	00	NOP
C :0x064B	00	NOP
C :0x064C	00	NOP
C :0x064D	00	NOP
C :0x064E	00	NOP
C :0x064F	00	NOP
C :0x0650	00	NOP
C :0x0651	00	NOP
C :0x0652	00	NOP
C :0x0653	00	NOP
C :0x0654	00	NOP
C :0x0655	00	NOP
C :0x0656	00	NOP
C :0x0657	00	NOP
C :0x0658	00	NOP
C :0x0659	00	NOP
C :0x065A	00	NOP
C :0x065B	00	NOP

C :0x065C	00	NOP
C :0x065D	00	NOP
C :0x065E	00	NOP
C :0x065F	00	NOP
C :0x0660	00	NOP
C :0x0661	00	NOP
C :0x0662	00	NOP
C :0x0663	00	NOP
C :0x0664	00	NOP
C :0x0665	00	NOP
C :0x0666	00	NOP
C :0x0667	00	NOP
C :0x0668	00	NOP
C :0x0669	00	NOP
C :0x066A	00	NOP
C :0x066B	00	NOP
C :0x066C	00	NOP
C :0x066D	00	NOP
C :0x066E	00	NOP
C :0x066F	00	NOP
C :0x0670	00	NOP
C :0x0671	00	NOP
C :0x0672	00	NOP
C :0x0673	00	NOP
C :0x0674	00	NOP
C :0x0675	00	NOP
C :0x0676	00	NOP
C :0x0677	00	NOP
C :0x0678	00	NOP
C :0x0679	00	NOP
C :0x067A	00	NOP
C :0x067B	00	NOP
C :0x067C	00	NOP
C :0x067D	00	NOP
C :0x067E	00	NOP
C :0x067F	00	NOP
C :0x0680	00	NOP
C :0x0681	00	NOP
C :0x0682	00	NOP
C :0x0683	00	NOP
C :0x0684	00	NOP
C :0x0685	00	NOP
C :0x0686	00	NOP
C :0x0687	00	NOP
C :0x0688	00	NOP
C :0x0689	00	NOP
C :0x068A	00	NOP
C :0x068B	00	NOP
C :0x068C	00	NOP
C :0x068D	00	NOP
C :0x068E	00	NOP
C :0x068F	00	NOP
C :0x0690	00	NOP
C :0x0691	00	NOP
C :0x0692	00	NOP
C :0x0693	00	NOP
C :0x0694	00	NOP
C :0x0695	00	NOP
C :0x0696	00	NOP
C :0x0697	00	NOP
C :0x0698	00	NOP
C :0x0699	00	NOP
C :0x069A	00	NOP
C :0x069B	00	NOP
C :0x069C	00	NOP
C :0x069D	00	NOP
C :0x069E	00	NOP
C :0x069F	00	NOP
C :0x06A0	00	NOP
C :0x06A1	00	NOP
C :0x06A2	00	NOP
C :0x06A3	00	NOP
C :0x06A4	00	NOP
C :0x06A5	00	NOP
C :0x06A6	00	NOP
C :0x06A7	00	NOP
C :0x06A8	00	NOP
C :0x06A9	00	NOP
C :0x06AA	00	NOP
C :0x06AB	00	NOP
C :0x06AC	00	NOP
C :0x06AD	00	NOP
C :0x06AE	00	NOP
C :0x06AF	00	NOP
C :0x06B0	00	NOP
C :0x06B1	00	NOP
C :0x06B2	00	NOP
C :0x06B3	00	NOP
C :0x06B4	00	NOP
C :0x06B5	00	NOP
C :0x06B6	00	NOP
C :0x06B7	00	NOP
C :0x06B8	00	NOP
C :0x06B9	00	NOP
C :0x06BA	00	NOP
C :0x06BB	00	NOP
C :0x06BC	00	NOP
C :0x06BD	00	NOP
C :0x06BE	00	NOP

C :0x06BF	00	NOP
C :0x06C0	00	NOP
C :0x06C1	00	NOP
C :0x06C2	00	NOP
C :0x06C3	00	NOP
C :0x06C4	00	NOP
C :0x06C5	00	NOP
C :0x06C6	00	NOP
C :0x06C7	00	NOP
C :0x06C8	00	NOP
C :0x06C9	00	NOP
C :0x06CA	00	NOP
C :0x06CB	00	NOP
C :0x06CC	00	NOP
C :0x06CD	00	NOP
C :0x06CE	00	NOP
C :0x06CF	00	NOP
C :0x06D0	00	NOP
C :0x06D1	00	NOP
C :0x06D2	00	NOP
C :0x06D3	00	NOP
C :0x06D4	00	NOP
C :0x06D5	00	NOP
C :0x06D6	00	NOP
C :0x06D7	00	NOP
C :0x06D8	00	NOP
C :0x06D9	00	NOP
C :0x06DA	00	NOP
C :0x06DB	00	NOP
C :0x06DC	00	NOP
C :0x06DD	00	NOP
C :0x06DE	00	NOP
C :0x06DF	00	NOP
C :0x06E0	00	NOP
C :0x06E1	00	NOP
C :0x06E2	00	NOP
C :0x06E3	00	NOP
C :0x06E4	00	NOP
C :0x06E5	00	NOP
C :0x06E6	00	NOP
C :0x06E7	00	NOP
C :0x06E8	00	NOP
C :0x06E9	00	NOP
C :0x06EA	00	NOP
C :0x06EB	00	NOP
C :0x06EC	00	NOP
C :0x06ED	00	NOP
C :0x06EE	00	NOP
C :0x06EF	00	NOP
C :0x06F0	00	NOP
C :0x06F1	00	NOP
C :0x06F2	00	NOP
C :0x06F3	00	NOP
C :0x06F4	00	NOP
C :0x06F5	00	NOP
C :0x06F6	00	NOP
C :0x06F7	00	NOP
C :0x06F8	00	NOP
C :0x06F9	00	NOP
C :0x06FA	00	NOP
C :0x06FB	00	NOP
C :0x06FC	00	NOP
C :0x06FD	00	NOP
C :0x06FE	00	NOP
C :0x06FF	00	NOP
C :0x0700	00	NOP
C :0x0701	00	NOP
C :0x0702	00	NOP
C :0x0703	00	NOP
C :0x0704	00	NOP
C :0x0705	00	NOP
C :0x0706	00	NOP
C :0x0707	00	NOP
C :0x0708	00	NOP
C :0x0709	00	NOP
C :0x070A	00	NOP
C :0x070B	00	NOP
C :0x070C	00	NOP
C :0x070D	00	NOP
C :0x070E	00	NOP
C :0x070F	00	NOP
C :0x0710	00	NOP
C :0x0711	00	NOP
C :0x0712	00	NOP
C :0x0713	00	NOP
C :0x0714	00	NOP
C :0x0715	00	NOP
C :0x0716	00	NOP
C :0x0717	00	NOP
C :0x0718	00	NOP
C :0x0719	00	NOP
C :0x071A	00	NOP
C :0x071B	00	NOP
C :0x071C	00	NOP
C :0x071D	00	NOP
C :0x071E	00	NOP
C :0x071F	00	NOP
C :0x0720	00	NOP
C :0x0721	00	NOP

C :0x0722	00	NOP
C :0x0723	00	NOP
C :0x0724	00	NOP
C :0x0725	00	NOP
C :0x0726	00	NOP
C :0x0727	00	NOP
C :0x0728	00	NOP
C :0x0729	00	NOP
C :0x072A	00	NOP
C :0x072B	00	NOP
C :0x072C	00	NOP
C :0x072D	00	NOP
C :0x072E	00	NOP
C :0x072F	00	NOP
C :0x0730	00	NOP
C :0x0731	00	NOP
C :0x0732	00	NOP
C :0x0733	00	NOP
C :0x0734	00	NOP
C :0x0735	00	NOP
C :0x0736	00	NOP
C :0x0737	00	NOP
C :0x0738	00	NOP
C :0x0739	00	NOP
C :0x073A	00	NOP
C :0x073B	00	NOP
C :0x073C	00	NOP
C :0x073D	00	NOP
C :0x073E	00	NOP
C :0x073F	00	NOP
C :0x0740	00	NOP
C :0x0741	00	NOP
C :0x0742	00	NOP
C :0x0743	00	NOP
C :0x0744	00	NOP
C :0x0745	00	NOP
C :0x0746	00	NOP
C :0x0747	00	NOP
C :0x0748	00	NOP
C :0x0749	00	NOP
C :0x074A	00	NOP
C :0x074B	00	NOP
C :0x074C	00	NOP
C :0x074D	00	NOP
C :0x074E	00	NOP
C :0x074F	00	NOP
C :0x0750	00	NOP
C :0x0751	00	NOP
C :0x0752	00	NOP
C :0x0753	00	NOP
C :0x0754	00	NOP
C :0x0755	00	NOP
C :0x0756	00	NOP
C :0x0757	00	NOP
C :0x0758	00	NOP
C :0x0759	00	NOP
C :0x075A	00	NOP
C :0x075B	00	NOP
C :0x075C	00	NOP
C :0x075D	00	NOP
C :0x075E	00	NOP
C :0x075F	00	NOP
C :0x0760	00	NOP
C :0x0761	00	NOP
C :0x0762	00	NOP
C :0x0763	00	NOP
C :0x0764	00	NOP
C :0x0765	00	NOP
C :0x0766	00	NOP
C :0x0767	00	NOP
C :0x0768	00	NOP
C :0x0769	00	NOP
C :0x076A	00	NOP
C :0x076B	00	NOP
C :0x076C	00	NOP
C :0x076D	00	NOP
C :0x076E	00	NOP
C :0x076F	00	NOP
C :0x0770	00	NOP
C :0x0771	00	NOP
C :0x0772	00	NOP
C :0x0773	00	NOP
C :0x0774	00	NOP
C :0x0775	00	NOP
C :0x0776	00	NOP
C :0x0777	00	NOP
C :0x0778	00	NOP
C :0x0779	00	NOP
C :0x077A	00	NOP
C :0x077B	00	NOP
C :0x077C	00	NOP
C :0x077D	00	NOP
C :0x077E	00	NOP
C :0x077F	00	NOP
C :0x0780	00	NOP
C :0x0781	00	NOP
C :0x0782	00	NOP
C :0x0783	00	NOP
C :0x0784	00	NOP

C :0x0785	00	NOP
C :0x0786	00	NOP
C :0x0787	00	NOP
C :0x0788	00	NOP
C :0x0789	00	NOP
C :0x078A	00	NOP
C :0x078B	00	NOP
C :0x078C	00	NOP
C :0x078D	00	NOP
C :0x078E	00	NOP
C :0x078F	00	NOP
C :0x0790	00	NOP
C :0x0791	00	NOP
C :0x0792	00	NOP
C :0x0793	00	NOP
C :0x0794	00	NOP
C :0x0795	00	NOP
C :0x0796	00	NOP
C :0x0797	00	NOP
C :0x0798	00	NOP
C :0x0799	00	NOP
C :0x079A	00	NOP
C :0x079B	00	NOP
C :0x079C	00	NOP
C :0x079D	00	NOP
C :0x079E	00	NOP
C :0x079F	00	NOP
C :0x07A0	00	NOP
C :0x07A1	00	NOP
C :0x07A2	00	NOP
C :0x07A3	00	NOP
C :0x07A4	00	NOP
C :0x07A5	00	NOP
C :0x07A6	00	NOP
C :0x07A7	00	NOP
C :0x07A8	00	NOP
C :0x07A9	00	NOP
C :0x07AA	00	NOP
C :0x07AB	00	NOP
C :0x07AC	00	NOP
C :0x07AD	00	NOP
C :0x07AE	00	NOP
C :0x07AF	00	NOP
C :0x07B0	00	NOP
C :0x07B1	00	NOP
C :0x07B2	00	NOP
C :0x07B3	00	NOP
C :0x07B4	00	NOP
C :0x07B5	00	NOP
C :0x07B6	00	NOP
C :0x07B7	00	NOP
C :0x07B8	00	NOP
C :0x07B9	00	NOP
C :0x07BA	00	NOP
C :0x07BB	00	NOP
C :0x07BC	00	NOP
C :0x07BD	00	NOP
C :0x07BE	00	NOP
C :0x07BF	00	NOP
C :0x07C0	00	NOP
C :0x07C1	00	NOP
C :0x07C2	00	NOP
C :0x07C3	00	NOP
C :0x07C4	00	NOP
C :0x07C5	00	NOP
C :0x07C6	00	NOP
C :0x07C7	00	NOP
C :0x07C8	00	NOP
C :0x07C9	00	NOP
C :0x07CA	00	NOP
C :0x07CB	00	NOP
C :0x07CC	00	NOP
C :0x07CD	00	NOP
C :0x07CE	00	NOP
C :0x07CF	00	NOP
C :0x07D0	00	NOP
C :0x07D1	00	NOP
C :0x07D2	00	NOP
C :0x07D3	00	NOP
C :0x07D4	00	NOP
C :0x07D5	00	NOP
C :0x07D6	00	NOP
C :0x07D7	00	NOP
C :0x07D8	00	NOP
C :0x07D9	00	NOP
C :0x07DA	00	NOP
C :0x07DB	00	NOP
C :0x07DC	00	NOP
C :0x07DD	00	NOP
C :0x07DE	00	NOP
C :0x07DF	00	NOP
C :0x07E0	00	NOP
C :0x07E1	00	NOP
C :0x07E2	00	NOP
C :0x07E3	00	NOP
C :0x07E4	00	NOP
C :0x07E5	00	NOP
C :0x07E6	00	NOP
C :0x07E7	00	NOP

```

C:0x07E8    00      NOP
C:0x07E9    00      NOP
C:0x07EA    00      NOP
C:0x07EB    00      NOP
C:0x07EC    00      NOP
C:0x07ED    00      NOP
C:0x07EE    00      NOP
C:0x07EF    00      NOP
C:0x07F0    00      NOP
C:0x07F1    00      NOP
C:0x07F2    00      NOP
C:0x07F3    00      NOP
C:0x07F4    00      NOP
C:0x07F5    00      NOP
C:0x07F6    00      NOP
C:0x07F7    00      NOP
C:0x07F8    00      NOP
C:0x07F9    00      NOP
C:0x07FA    00      NOP
C:0x07FB    00      NOP
C:0x07FC    00      NOP
C:0x07FD    00      NOP
C:0x07FE    00      NOP
C:0x07FF    00      NOP

71: void main (void){
72:     // Start of peripheral setup
73:     U8 TCON_save;
74:     // Watchdog
75:             WDTCN = 0xDE;                                // First key
C:0x0800    7597DE    MOV     WDTCN(0x97),#0xDE
76:             WDTCN = 0xAD;                                // Second key - Watchdog now disabled
77:     // Clock
C:0x0803    7597AD    MOV     WDTCN(0x97),#0xAD
78:             CLKSEL =
79:                     CLKSEL_CLKSL__HFOSC          | // Use 24.5MHz interal clock
80:                     CLKSEL_CLKDIV__SYSCLK_DIV_1; // Do not divide
81:     // Port 0
C:0x0806    E4        CLR     A
C:0x0807    F5A9      MOV     CLKSEL(0xA9),A
82:             POMDOUT =
83:                     POMDOUT_BO__PUSH_PULL      | // PWM1 output
84:                     POMDOUT_B1__PUSH_PULL      | // PWM2 output
85:                     POMDOUT_B2__OPEN_DRAIN
86:                     POMDOUT_B3__OPEN_DRAIN
87:                     POMDOUT_B4__PUSH_PULL      | // UART TX
88:                     POMDOUT_B5__OPEN_DRAIN      | // UART RX
89:                     POMDOUT_B6__OPEN_DRAIN
90:                     POMDOUT_B7__OPEN_DRAIN;
91:     // Port 1
C:0x0809    75A413    MOV     POMDOUT(0xA4),#0x13
92:             P1MDOUT =
93:                     P1MDOUT_BO__OPEN_DRAIN
94:                     P1MDOUT_B1__OPEN_DRAIN
95:                     P1MDOUT_B2__OPEN_DRAIN
96:                     P1MDOUT_B3__PUSH_PULL      | // TEST2
97:                     P1MDOUT_B4__PUSH_PULL      | // TEST1
98:                     P1MDOUT_B5__OPEN_DRAIN
99:                     P1MDOUT_B6__OPEN_DRAIN
100:                    P1MDOUT_B7__OPEN_DRAIN;
C:0x080C    75A518    MOV     P1MDOUT(0xA5),#0x18
101:            P1MDIN =
102:                    P1MDIN_BO__ANALOG
| 103:        // ADC1
| 104:        P1MDIN_B1__ANALOG
| 105:        // ADC2
| 106:        P1MDIN_B2__ANALOG
| 107:        // ADC3
| 108:        P1MDIN_B3__DIGITAL
| 109:        P1MDIN_B4__DIGITAL
| 110:        P1MDIN_B5__DIGITAL
| 111:        P1MDIN_B6__DIGITAL
| 112:        P1MDIN_B7__DIGITAL;
110:    // Port crossbar
C:0x080F    75F2F8    MOV     P1MDIN(0xF2),#0x8
111:            XBRO =
112:                    XBRO_UTTOE__ENABLED      | // Route out UART
113:                    XBRO_SPIOE__DISABLED
114:                    XBRO_SMBOE__DISABLED
115:                    XBRO_CPOE__DISABLED
116:                    XBRO_CPOAE__DISABLED
117:                    XBRO_CP1E__DISABLED
|
118:                    XBRO_CPIAE__DISABLED
119:                    XBRO_SYSCKE__DISABLED;
C:0x0812    75E101    MOV     XBRO(0xE1),#0x01
120:            XBR1 =
121:                    XBR1_PCAOME__CEX0_CEX1
122:                    XBR1_ECIE__DISABLED
123:                    XBR1_TOE__DISABLED
|
124:                    XBR1_T1E__DISABLED
|
125:                    XBR1_T2E__DISABLED;
C:0x0815    75E202    MOV     XBR1(0xE2),#0x02
126:            XBR2 =

```

```

127:                               XBR2_WEAKPUD__PULL_UPS_ENABLED   |      // Weak pull ups
128:                               XBR2_XBARE__ENABLED;           |      // Enable cross bar
129:                               // ADC
130:                               //ADCOMX =
// Mux set in application
131:                               //      ADCMX_ADCOMX__ADCOP10;
C:0x0818    75E340    MOV      XBR2(0xE3),#0x40
132:                               ADCOCF =
133:                               (1 << ADCOCF_ADSC__SHIFT)
|
134:                               ADCOCF_AD8BE__NORMAL          |      // ADC set to 10 bit
135:                               ADCOCF_ADGN__GAIN_1           |      // ADC gain set to 1
136:                               ADCOCF_ADTM__TRACK_NORMAL;
// Immediate covert
C:0x081B    75BC09    MOV      ADCOCF(0xBC),#0x09
137:                               ADCOCNO &=
138:                               ~ADCOCON_ADCM__FMASK;
C:0x081E    53E8F8    ANL      ADCOCNO(0xE8),#0xF8
139:                               ADCOCNO |=
140:                               ADCOCNO_ADEN__ENABLED
141:                               ADCOCNO_ADCM__ADBUSD;
142:                               // Timer 0
C:0x0821    43E880    ORL      ADCOCNO(0xE8),#0x80
143:                               TCON_save = TCON;
C:0x0824    AF88      MOV      R7,TCON(0x88)
144:                               TCON &=
145:                               TCON_TRO__BMASK
&
146:                               TCON_TR1__BMASK;
C:0x0826    F588      MOV      TCON(0x88),A
147:                               TH1 = (150 << TH1_TH1__SHIFT);
C:0x0828    758D96    MOV      TH1(0x8D),#0x96
148:                               TL1 = (150 << TL1_TL1__SHIFT);
C:0x082B    758B96    MOV      TL1(0x8B),#0x96
149:                               TCON =
150:                               TCON_save;
151:                               // Timer setup
C:0x082E    8F88      MOV      TCON(0x88),R7
152:                               CKCON =
153:                               CKCON_SCA__SYSCLK_DIV_12
|
154:                               CKCON_TOM__PRESCALE
155:                               CKCON_T3MH__EXTERNAL_CLOCK
156:                               CKCON_T3ML__EXTERNAL_CLOCK
|
157:                               CKCON_TIM__SYSCLK;
C:0x0830    758E08    MOV      CKCON(0x8E),#0x08
158:                               TMOD =
159:                               TMOD_TOM__MODE0
|
160:                               TMOD_CTO__TIMER
|
161:                               TMOD_GATE0__DISABLED
162:                               TMOD_TIM__MODE2
|
163:                               TMOD_CT1__TIMER
|
164:                               TMOD_GATE1__DISABLED;
C:0x0833    758920    MOV      TMOD(0x89),#0x20
165:                               TCON |=
166:                               TCON_TR1__RUN ;
167:                               // UART
C:0x0836    438840    ORL      TCON(0x88),#0x40
168:                               SCONO |=
169:                               SCONO_REN__RECEIVE_ENABLED;
170:                               // VREF
C:0x0839    439810    ORL      SCONO(0x98),#0x10
171:                               REFOCN =
172:                               REFOCN_REFSL__VDD_PIN
173:                               REFOCN_IREFLVL__1P65
174:                               REFOCN_GNDSL__GND_PIN
175:                               REFOCN_TEMPE__TEMP_DISABLED;
176:                               // RSTSRC
C:0x083C    75D108    MOV      REFOCN(0xD1),#0x08
177:                               RSTSRC =
178:                               RSTSRC_CORSEF__NOT_SET
179:                               RSTSRC_MCDRSF__SET
|
180:                               RSTSRC_PORSF__SET
|
181:                               RSTSRC_SWRSF__NOT_SET;
182:                               // Interrupt
C:0x083F    75EF06    MOV      RSTSRC(0xEF),#0x06
183:                               IE =
184:                               IE_EA__ENABLED
185:                               IE_EX0__DISABLED
|
186:                               IE_EX1__DISABLED
|
187:                               IE_ESPIO__DISABLED
|
188:                               IE_ET0__DISABLED
|
189:                               IE_ET1__ENABLED
190:                               IE_ET2__ENABLED
|
191:                               IE_ESO__ENABLED;
C:0x0842    75A8B8    MOV      IE(0xA8),#IP(0xB8)
192:                               IP =

```

```

193:           IP_PX0__LOW          |
194:           IP_PX1__LOW          |
195:           IP_PSPIO__LOW        |
196:           IP_PTO__LOW          |
197:           IP_PT1__LOW          |
|           IP_PT2__HIGH         | // Timer 2 highest priority
198:           IP_PSO__LOW          |
199:           IP_PSO__LOW          |
200:           // PCA
C:0x0845  75B820  MOV     IP(0xB8),#0x20
201:           PCAOCN_CR =      PCAOCN_CR_STOP;
202:           PCAOCN_CR_STOP;
C:0x0848  C2DE   CLR     PCAOCN_CR(0xD8.6)
203:           PCAOCPMO =       PCAOCPMO_CAPN_DISABLED
204:           PCAOCPMO_CAPN_DISABLED
|           PCAOCPMO_ECCF_ENABLED    |
205:           PCAOCPMO_MAT_ENABLED   |
206:           PCAOCPMO_CAPP_DISABLED |
207:           PCAOCPMO_CAPP_DISABLED |
|           PCAOCPMO_ECOM_ENABLED   |
208:           PCAOCPMO_PWM_ENABLED   |
209:           PCAOCPMO_TOG_DISABLED  |
210:           PCAOCPMO_TOG_DISABLED  |
C:0x084A  75DA4B  MOV     PCAOCPMO(0xDA),#0x4B
211:           PCAOCPM1 =       PCAOCPM1_CAPN_DISABLED
212:           PCAOCPM1_CAPN_DISABLED
|           PCAOCPM1_ECCF_ENABLED   |
213:           PCAOCPM1_MAT_ENABLED   |
214:           PCAOCPM1_CAPP_DISABLED |
215:           PCAOCPM1_CAPP_DISABLED |
|           PCAOCPM1_ECOM_ENABLED   |
216:           PCAOCPM1_PWM_ENABLED   |
217:           PCAOCPM1_TOG_DISABLED  |
218:           PCAOCPM1_TOG_DISABLED  |
C:0x084D  75DB4B  MOV     PCAOCPM1(0xDB),#0x4B
219:           PCAOMD =       PCAOMD_CIDL_NORMAL
220:           PCAOMD_CIDL_NORMAL
221:           PCAOMD_ECF_OVF_INT_DISABLED |
222:           PCAOMD_CPS_SYSCLK;
C:0x0850  75D908  MOV     PCAOMD(0xD9),#0x08
223:           PCAOCN |=      PCAOCN_CR_RUN;
224:           PCAOCN_CR_RUN;
225:           // Timer 2
C:0x0853  43D840  ORL     PCAOCN(0xD8),#0x40
226:           TMR2CN |= TMR2CN_TR2_RUN;
227:           // End of peripheral setup
228:
C:0x0856  43C804  ORL     TMR2CN(0xC8),#0x04
229:           enabled = false;
C:0x0859  C202   CLR     enabled(0x20.2)
230:           integral = 0;
C:0x085B  F52E   MOV     integral(0x2E),A
C:0x085D  F52F   MOV     0x2F,A
231:           target_mv = DEFAULT_OUT_MV;
C:0x085F  752513  MOV     target_mv(0x25),#0x13
C:0x0862  752688  MOV     0x26,#TCUN(0x88)
232:           high_mv = DEFAULT_HIGH_MV;
C:0x0865  752717  MOV     high_mv(0x27),#0x17
C:0x0868  752870  MOV     0x28,#0x70
233:           low_mv = DEFAULT_LOW_MV;
234:
C:0x086B  75370F  MOV     low_mv(0x37),#0x0F
C:0x086E  7538A0  MOV     0x38,#0xA0
235:           SCON0_TI = 1;
C:0x0871  D299   SETB   SCON0_TI(0x98.1)
236:           SCON0_RI = 0;
237:
C:0x0873  C298   CLR     SCON0_RI(0x98.0)
238:           while (1){
239:               if(SCON0_RI){
C:0x0875  3098FD  JNB     SCON0_RI(0x98.0),C:0875
240:               SCON0_RI = 0;
C:0x0878  C298   CLR     SCON0_RI(0x98.0)
241:               uart_in[4] = uart_in[3];
// Small buffer, less code not to use loop
C:0x087A  853536  MOV     0x36,0x35
242:               uart_in[3] = uart_in[2];
C:0x087D  853435  MOV     0x35,0x34
243:               uart_in[2] = uart_in[1];
C:0x0880  853334  MOV     0x34,0x33
244:               uart_in[1] = uart_in[0];
C:0x0883  853233  MOV     0x33,uart_in(0x32)
245:               uart_in[0] = SBUFO;
246:
C:0x0886  859932  MOV     uart_in(0x32),SBUFO(0x99)
247:               uartLoadOut(uart_in[0]);
248:
C:0x0889  AF32   MOV     R7,uart_in(0x32)
C:0x088B  120B9C  LCALL  uartLoadOut(C:0B9C)
249:               switch(uart_in[0]){
C:0x088E  E532   MOV     A,uart_in(0x32)
C:0x0890  120AA6  LCALL  C?CCASE(C:0AA6)
C:0x0893  08     INC     R0
C:0x0894  B56108  CJNE   A,0x61,C:089F
C:0x0897  BD6308  CJNE   R5,#0x63,C:08A2
C:0x089A  C564   XCH    A,0x64
C:0x089C  08     INC     R0
C:0x089D  CD     XCH    A,R5
C:0x089E  67     XRL    A,@R1

```

```

C:0x089F    08      INC     R0
C:0x08A0    DA6D    DJNZ    R2,C:090F
C:0x08A2    08      INC     R0
C:0x08A3    E2      MOVX   A,@R0
C:0x08A4    6E      XRL    A,R6
C:0x08A5    08      INC     R0
C:0x08A6    D6      XCHD   A,@R0
C:0x08A7    73      JMP    @A+DPTR
C:0x08A8    08      INC     R0
C:0x08A9    EA      MOV    A,R2
C:0x08AA    7808   MOV    R0,#0x08
C:0x08AC    F2      MOVX   @R0,A
C:0x08AD    7908   MOV    R1,#0x08
C:0x08AF    FA      MOV    R2,A
C:0x08B0    7A00   MOV    R2,#0x00
C:0x08B2    00      NOP
C:0x08B3    09      INC     R1
C:0x08B4    06      INC     @R0
250:           case 'a':   uartNumbers(target_mV,true);
C:0x08B5    D200   SETB   0x20.0
C:0x08B7    AF26   MOV    R7,0x26
C:0x08B9    AE25   MOV    R6,target_mV(0x25)
251:           break;
C:0x08BB    8043   SJMP   C:0900
252:           case 'c':   uartNumbers(current,true);
C:0x08BD    D200   SETB   0x20.0
C:0x08BF    AF22   MOV    R7,0x22
C:0x08C1    AE21   MOV    R6,current(0x21)
253:           break;
C:0x08C3    803B   SJMP   C:0900
254:           case 'd':   uartNumbers(duty,true);
C:0x08C5    AF29   MOV    R7,duty(0x29)
C:0x08C7    7E00   MOV    R6,#0x00
C:0x08C9    D200   SETB   0x20.0
255:           break;
C:0x08CB    8033   SJMP   C:0900
256:           case 'g':   enabled = true;
C:0x08CD    D202   SETB   enabled(0x20.2)
257:           integral
= 0;
C:0x08CF    E4      CLR    A
C:0x08D0    F52E   MOV    integral(0xE),A
C:0x08D2    F52F   MOV    0xF,A
258:           break;
C:0x08D4    809F   SJMP   C:0875
259:           case 's':   enabled = false;
C:0x08D6    C202   CLR    enabled(0x20.2)
260:           break;
C:0x08D8    809B   SJMP   C:0875
261:           case 'm':   uartNumbers(high_mV,true);
C:0x08DA    D200   SETB   0x20.0
C:0x08DC    AF28   MOV    R7,0x28
C:0x08DE    AE27   MOV    R6,high_mV(0x27)
262:           break;
C:0x08E0    801E   SJMP   C:0900
263:           case 'n':   uartNumbers(low_mV,true);
C:0x08E2    D200   SETB   0x20.0
C:0x08E4    AF38   MOV    R7,0x38
C:0x08E6    AE37   MOV    R6,low_mV(0x37)
264:           break;
C:0x08E8    8016   SJMP   C:0900
265:           case 'x':   uartNumbers(adcl,true);
C:0x08EA    D200   SETB   0x20.0
C:0x08EC    AF2B   MOV    R7,0x2B
C:0x08EE    AE2A   MOV    R6,adcl(0x2A)
266:           break;
C:0x08F0    800E   SJMP   C:0900
267:           case 'y':   uartNumbers(adc2,true);
C:0x08F2    D200   SETB   0x20.0
C:0x08F4    AF2D   MOV    R7,0x2D
C:0x08F6    AE2C   MOV    R6,adc2(0x2C)
268:           break;
C:0x08F8    8006   SJMP   C:0900
269:           case 'z':   uartNumbers(adc3,true);
C:0x08FA    D200   SETB   0x20.0
C:0x08FC    AF31   MOV    R7,0x31
C:0x08FE    AE30   MOV    R6,adc3(0x30)
C:0x0900    120AF9 LCALL  uartNumbers(C:0AF9)
270:           break;
C:0x0903    020875 LJMP   C:0875
271:           default:  switch(uart_in[4]){
C:0x0906    E536   MOV    A,0x36
C:0x0908    2494   ADD    A,#0x94
C:0x090A    601B   JZ     C:0927
C:0x090C    24F6   ADD    A,#0xF6
C:0x090E    6027   JZ     C:0937
C:0x0910    240E   ADD    A,#0x0E
C:0x0912    6003   JZ     C:0917
C:0x0914    020875 LJMP   C:0875
272:           case 'h':
high_mV        = uartNumbers(high_mV,false);
C:0x0917    C200   CLR    0x20.0
C:0x0919    AF28   MOV    R7,0x28
C:0x091B    AE27   MOV    R6,high_mV(0x27)
C:0x091D    120AF9 LCALL  uartNumbers(C:0AF9)
C:0x0920    8E27   MOV    high_mV(0x27),R6
C:0x0922    8F28   MOV    0x28,R7
273:           break;

```

```

C:0x0924    020875   LJMP    C:0875
274:
low_mV = uartNumbers(low_mV, false);
C:0x0927    C200     CLR     0x20.0
C:0x0929    AF38     MOV     R7, 0x38
C:0x092B    AE37     MOV     R6, low_mV(0x37)
C:0x092D    120AF9   LCALL   uartNumbers(C:0AF9)
C:0x0930    8E37     MOV     low_mV(0x37), R6
C:0x0932    8F38     MOV     0x38, R7
275:
break;
C:0x0934    020875   LJMP    C:0875
276:
target_mV = uartNumbers(target_mV, false);
C:0x0937    C200     CLR     0x20.0
C:0x0939    AF26     MOV     R7, 0x26
C:0x093B    AE25     MOV     R6, target_mV(0x25)
C:0x093D    120AF9   LCALL   uartNumbers(C:0AF9)
C:0x0940    8E25     MOV     target_mV(0x25), R6
C:0x0942    8F26     MOV     0x26, R7
277:
integral = 0;
C:0x0944    E4       CLR     A
C:0x0945    F52E     MOV     integral(0xE), A
C:0x0947    F52F     MOV     0x2F, A
278:
break;
C:0x0949    020875   LJMP    C:0875
341: INTERRUPT (TIMER2_ISR, TIMER2_IRQn){ // One timer handling UART Tx and PID
342:     int out;
343:     int error;
C:0x094C    COEO     PUSH    0xE0
C:0x094E    COFO     PUSH    0xF0
C:0x0950    COD0     PUSH    0xD0
C:0x0952    75D000   MOV     0xD0, #0x00
C:0x0955    C000     PUSH    0x00
C:0x0957    C004     PUSH    0x04
C:0x0959    C005     PUSH    0x05
C:0x095B    C006     PUSH    0x06
C:0x095D    C007     PUSH    0x07
344:     TEST1 = 1;
345:
346:
C:0x095F    D294     SETB    TEST1(0x90.4)
347:     adc1 = readAdc(ADC1);
C:0x0961    7F08     MOV     R7, #0x08
C:0x0963    120B7B   LCALL   readAdc(C:0B7B)
C:0x0966    8E2A     MOV     adc1(0x2A), R6
C:0x0968    8F2B     MOV     0x2B, R7
348:     adc2 = readAdc(ADC2);
C:0x096A    7F09     MOV     R7, #0x09
C:0x096C    120B7B   LCALL   readAdc(C:0B7B)
C:0x096F    8E2C     MOV     adc2(0x2C), R6
C:0x0971    8F2D     MOV     0x2D, R7
349:     adc3 = readAdc(ADC3);
C:0x0973    7F0A     MOV     R7, #0x0A
C:0x0975    120B7B   LCALL   readAdc(C:0B7B)
C:0x0978    8E30     MOV     adc3(0x30), R6
C:0x097A    8F31     MOV     0x31, R7
350:     current = (adc2 - adc3)*10;
351:
C:0x097C    C3       CLR     C
C:0x097D    E52D     MOV     A, 0x2D
C:0x097F    9531     SUBB   A, 0x31
C:0x0981    FF       MOV     R7, A
C:0x0982    E52C     MOV     A, adc2(0x2C)
C:0x0984    9530     SUBB   A, adc3(0x30)
C:0x0986    FE       MOV     R6, A
C:0x0987    7C00     MOV     R4, #0x00
C:0x0989    7D0A     MOV     R5, #0xA
C:0x098B    120A2C   LCALL   C?IMUL(C:0A2C)
C:0x098E    8E21     MOV     current(0x21), R6
C:0x0990    8F22     MOV     0x22, R7
352:     if((adc1 < low_mV) || (adc1 > high_mV)){ // Watch input voltage
C:0x0992    C3       CLR     C
C:0x0993    E52B     MOV     A, 0x2B
C:0x0995    9538     SUBB   A, 0x38
C:0x0997    E52A     MOV     A, adc1(0x2A)
C:0x0999    9537     SUBB   A, low_mV(0x37)
C:0x099B    400B     JC      C:09A8
C:0x099D    D3       SETB   C
C:0x099E    E52B     MOV     A, 0x2B
C:0x09A0    9528     SUBB   A, 0x28
C:0x09A2    E52A     MOV     A, adc1(0x2A)
C:0x09A4    9527     SUBB   A, high_mV(0x27)
C:0x09A6    4002     JC      C:09AA
353:     enabled = false;
C:0x09A8    C202     CLR     enabled(0x20.2)
354: }
355:
356:
357:     error = (int)target_mV - (int)adc3; // PID controller
C:0x09AA    AE30     MOV     R6, adc3(0x30)
C:0x09AC    AF31     MOV     R7, 0x31
C:0x09AE    AC25     MOV     R4, target_mV(0x25)
C:0x09B0    AD26     MOV     R5, 0x26
C:0x09B2    C3       CLR     C
C:0x09B3    ED       MOV     A, R5
C:0x09B4    9F       SUBB   A, R7

```

```

C:0x09B5    FF      MOV     R7,A
C:0x09B6    EC      MOV     A,R4
C:0x09B7    9E      SUBB   A,R6
C:0x09B8    FE      MOV     R6,A
358:        integral += error;
C:0x09B9    EF      MOV     A,R7
C:0x09BA    252F    ADD     A,0x2F
C:0x09BC    F52F    MOV     0x2F,A
C:0x09BE    EE      MOV     A,R6
C:0x09BF    352E    ADDC   A,integral(0x2E)
C:0x09C1    F52E    MOV     integral(0x2E),A
359:        out = ((error*I) + (integral*I)) >> 10; // Divide by 1024
C:0x09C3    7C00    MOV     R4,#0x00
C:0x09C5    7D05    MOV     R5,#0x05
C:0x09C7    120A2C  LCALL  C?IMUL(C:0A2C)
C:0x09CA    AC06    MOV     R4,0x06
C:0x09CE    AD07    MOV     R5,0x07
C:0x09CC    E52F    MOV     A,0x2F
C:0x09D0    25E0    ADD     A,0xE0
C:0x09D2    FF      MOV     R7,A
C:0x09D3    E52E    MOV     A,integral(0x2E)
C:0x09D5    33      RLC     A
C:0x09D6    FE      MOV     R6,A
C:0x09D7    ED      MOV     A,R5
C:0x09D8    2F      ADD     A,R7
C:0x09D9    FF      MOV     R7,A
C:0x09DA    EC      MOV     A,R4
C:0x09DB    3E      ADDC   A,R6
C:0x09DC    FE      MOV     R6,A
C:0x09DD    EF      MOV     A,R7
C:0x09DE    780A    MOV     R0,#0x0A
C:0x09E0    CE      XCH     A,R6
C:0x09E1    A2E7    MOV     C,0xE0.7
C:0x09E3    13      RRC     A
C:0x09E4    CE      XCH     A,R6
C:0x09E5    13      RRC     A
C:0x09E6    D8F8    DJNZ   R0,C:09E0
C:0x09E8    FF      MOV     R7,A
360:        if((out < 0) || (!enabled)){
C:0x09E9    C3      CLR     C
C:0x09EA    EE      MOV     A,R6
C:0x09EB    6480    XRL     A,#0x80
C:0x09ED    9480    SUBB   A,#0x80
C:0x09EF    4003    JC     C:09F4
C:0x09F1    200202  JB     enabled(0x20.2),C:09F6
361:        out = 0;
C:0x09F4    7FO0    MOV     R7,#0x00
362:    }
363:        duty = out;
C:0x09F6    8F29    MOV     duty(0x29),R7
364:        PCAOCPH0 = PCAOCPH1 = 0xFF - out;
365:
366:
367:
C:0x09F8    C3      CLR     C
C:0x09F9    74FF    MOV     A,#0xF
C:0x09FB    9F      SUBB   A,R7
C:0x09FC    F5EA    MOV     PCAOCPH1(0xEA),A
C:0x09FE    F5FC    MOV     PCAOCPH0(0xFC),A
368:        if(head != tail){
C:0x0A00    E523    MOV     A,head(0x23)
C:0x0A02    6524    XRL     A,tail(0x24)
C:0x0A04    600E    JZ     C:0A14
369:        SBUFO = uart_out[tail]; // Timer tuned so no need to check
C:0x0A06    AF24    MOV     R7,tail(0x24)
C:0x0A08    7439    MOV     A,#uart_out(0x39)
C:0x0A0A    2F      ADD     A,R7
C:0x0A0B    F8      MOV     R0,A
C:0x0A0C    E6      MOV     A,@R0
C:0x0A0D    F599    MOV     SBUFO(0x99),A
370:        tail++; // Transmit UART
C:0x0A0F    0524    INC     tail(0x24)
371:        tail %= UART_SIZE_OUT; // Wrap around
C:0x0A11    532407  ANL     tail(0x24),#0x07
372:
373:
374:
375:        TMR2H = 255; // Runs at 4KHz
C:0x0A14    75C0FF  MOV     TMR2H(0xCD),#0xFF
376:        TMR2CN_TF2H = 0;
// Enable interrupt again
C:0x0A17    C2CF    CLR     TMR2CN_TF2H(0xC8.7)
377:        TEST1 = 0;
// Timing debug
C:0x0A19    C294    CLR     TEST1(0x90.4)
378:    }
C:0x0A1B    D007    POP     0x07
C:0x0A1D    D006    POP     0x06
C:0x0A1F    D005    POP     0x05
C:0x0A21    D004    POP     0x04
C:0x0A23    D000    POP     0x00
C:0x0A25    D0D0    POP     0xD0
C:0x0A27    D0F0    POP     0xF0
C:0x0A29    D0E0    POP     0xE0
C:0x0A2B    32      RETI
C?IMUL:
C:0x0A2C    EF      MOV     A,R7
C:0x0A2D    8DF0    MOV     0xF0,R5
C:0x0A2F    A4      MUL     AB

```

```

C:0x0A30    A8F0    MOV     R0 ,0xF0
C:0x0A32    CF      XCH     A ,R7
C:0x0A33    8CF0    MOV     0xF0 ,R4
C:0x0A35    A4      MUL     AB
C:0x0A36    28      ADD     A ,R0
C:0x0A37    CE      XCH     A ,R6
C:0x0A38    8DFO    MOV     0xF0 ,R5
C:0x0A3A    A4      MUL     AB
C:0x0A3B    2E      ADD     A ,R6
C:0x0A3C    FE      MOV     R6 ,A
C:0x0A3D    22      RET

C?UIDIV:
C:0x0A3E    BC000B  CJNE   R4 ,#0x00 ,C:0A4C
C:0x0A41    BE0029  CJNE   R6 ,#0x00 ,C:0A6D
C:0x0A44    EF      MOV     A ,R7
C:0x0A45    8DFO    MOV     0xF0 ,R5
C:0x0A47    84      DIV     AB
C:0x0A48    FF      MOV     R7 ,A
C:0x0A49    ADFO    MOV     R5 ,0xF0
C:0x0A4B    22      RET
C:0x0A4C    E4      CLR     A
C:0x0A4D    CC      XCH     A ,R4
C:0x0A4E    F8      MOV     R0 ,A
C:0x0A4F    75F008  MOV     0xF0 ,#0x08
C:0x0A52    EF      MOV     A ,R7
C:0x0A53    2F      ADD     A ,R7
C:0x0A54    FF      MOV     R7 ,A
C:0x0A55    EE      MOV     A ,R6
C:0x0A56    33      RLC     A
C:0x0A57    FE      MOV     R6 ,A
C:0x0A58    EC      MOV     A ,R4
C:0x0A59    33      RLC     A
C:0x0A5A    FC      MOV     R4 ,A
C:0x0A5B    EE      MOV     A ,R6
C:0x0A5C    9D      SUBB   A ,R5
C:0x0A5D    EC      MOV     A ,R4
C:0x0A5E    98      SUBB   A ,R0
C:0x0A5F    4005    JC     C:0A66
C:0x0A61    FC      MOV     R4 ,A
C:0x0A62    EE      MOV     A ,R6
C:0x0A63    9D      SUBB   A ,R5
C:0x0A64    FE      MOV     R6 ,A
C:0x0A65    OF      INC     R7
C:0x0A66    D5F0E9  DJNZ   0xF0 ,C:0A52
C:0x0A69    E4      CLR     A
C:0x0A6A    CE      XCH     A ,R6
C:0x0A6B    FD      MOV     R5 ,A
C:0x0A6C    22      RET
C:0x0A6D    ED      MOV     A ,R5
C:0x0A6E    F8      MOV     R0 ,A
C:0x0A6F    F5FO    MOV     0xF0 ,A
C:0x0A71    EE      MOV     A ,R6
C:0x0A72    84      DIV     AB
C:0x0A73    20D21C  JB     0xD0 .2 ,C:0A92
C:0x0A76    FE      MOV     R6 ,A
C:0x0A77    ADFO    MOV     R5 ,0xF0
C:0x0A79    75F008  MOV     0xF0 ,#0x08
C:0x0A7C    EF      MOV     A ,R7
C:0x0A7D    2F      ADD     A ,R7
C:0x0A7E    FF      MOV     R7 ,A
C:0x0A7F    ED      MOV     A ,R5
C:0x0A80    33      RLC     A
C:0x0A81    FD      MOV     R5 ,A
C:0x0A82    4007    JC     C:0A8B
C:0x0A84    98      SUBB   A ,R0
C:0x0A85    5006    JNC    C:0A8D
C:0x0A87    D5F0F2  DJNZ   0xF0 ,C:0A7C
C:0x0A8A    22      RET
C:0x0A8B    C3      CLR     C
C:0x0A8C    98      SUBB   A ,R0
C:0x0A8D    FD      MOV     R5 ,A
C:0x0A8E    OF      INC     R7
C:0x0A8F    D5FOEA  DJNZ   0xF0 ,C:0A7C
C:0x0A92    22      RET
C?ULSHR:
C:0x0A93    E8      MOV     A ,R0
C:0x0A94    600F    JZ     C:0AA5
C:0x0A96    EC      MOV     A ,R4
C:0x0A97    C3      CLR     C
C:0x0A98    13      RRC     A
C:0x0A99    FC      MOV     R4 ,A
C:0x0A9A    ED      MOV     A ,R5
C:0x0A9B    13      RRC     A
C:0x0A9C    FD      MOV     R5 ,A
C:0x0A9D    EE      MOV     A ,R6
C:0x0A9E    13      RRC     A
C:0x0A9F    FE      MOV     R6 ,A
C:0x0AA0    EF      MOV     A ,R7
C:0x0AA1    13      RRC     A
C:0x0AA2    FF      MOV     R7 ,A
C:0x0AA3    D8F1    DJNZ   R0 ,C:0A96
C:0x0AA5    22      RET
C?CCASE:
C:0x0AA6    D083    POP    0x83
C:0x0AA8    D082    POP    0x82
C:0x0AAA    F8      MOV     R0 ,A
C:0x0AAB    E4      CLR     A
C:0x0AAC    93      MOVC   A ,@A+DPTR
C:0x0AAD    7012    JNZ    C:0AC1

```

```

C:0x0AAF    7401    MOV     A,#0x01
C:0x0AB1    93       MOVC    A,@A+DPTR
C:0x0AB2    700D    JNZ    C:0AC1
C:0x0AB4    A3       INC    DPTR
C:0x0AB5    A3       INC    DPTR
C:0x0AB6    93       MOVC    A,@A+DPTR
C:0x0AB7    F8       MOV    R0,A
C:0x0AB8    7401    MOV     A,#0x01
C:0x0ABA    93       MOVC    A,@A+DPTR
C:0x0ABB    F582    MOV    0x82,A
C:0x0ABD    8883    MOV    0x83,R0
C:0x0ABF    E4       CLR    A
C:0x0AC0    73       JMP    @A+DPTR
C:0x0AC1    7402    MOV     A,#0x02
C:0x0AC3    93       MOVC    A,@A+DPTR
C:0x0AC4    68       XRL    A,R0
C:0x0AC5    60EF    JZ     C:0AB6
C:0x0AC7    A3       INC    DPTR
C:0x0AC8    A3       INC    DPTR
C:0x0AC9    A3       INC    DPTR
C:0x0ACA    80DF    SJMP   C:0AAB
C?LIMUL:
C:0x0ACC    EC       MOV    A,R4
C:0x0ACD    8EFO    MOV    0xF0,R6
C:0x0ACF    A4       MUL    AB
C:0x0ADO    CC       XCH    A,R4
C:0x0AD1    C5F0    XCH    A,0xF0
C:0x0AD3    CC       XCH    A,R4
C:0x0AD4    CD       XCH    A,R5
C:0x0AD5    F8       MOV    R0,A
C:0x0AD6    EF       MOV    A,R7
C:0x0AD7    A4       MUL    AB
C:0x0AD8    CE       XCH    A,R6
C:0x0AD9    C5F0    XCH    A,0xF0
C:0x0ADB    2D       ADD    A,R5
C:0x0ADC    FD       MOV    R5,A
C:0x0ADD    E4       CLR    A
C:0x0ADE    3C       ADDC   A,R4
C:0x0ADF    FC       MOV    R4,A
C:0x0AE0    E8       MOV    A,R0
C:0x0AE1    A4       MUL    AB
C:0x0AE2    2E       ADD    A,R6
C:0x0AE3    C8       XCH    A,R0
C:0x0AE4    C5F0    XCH    A,0xF0
C:0x0AE6    3D       ADDC   A,R5
C:0x0AE7    FD       MOV    R5,A
C:0x0AE8    E4       CLR    A
C:0x0AE9    3C       ADDC   A,R4
C:0x0AEA    FC       MOV    R4,A
C:0x0AEB    EF       MOV    A,R7
C:0x0AEC    A4       MUL    AB
C:0x0AED    FF       MOV    R7,A
C:0x0AEE    E5F0    MOV    A,0xF0
C:0x0AFO    28       ADD    A,R0
C:0x0AF1    FE       MOV    R6,A
C:0x0AF2    E4       CLR    A
C:0x0AF3    3D       ADDC   A,R5
C:0x0AF4    FD       MOV    R5,A
C:0x0AF5    E4       CLR    A
C:0x0AF6    3C       ADDC   A,R4
C:0x0AF7    FC       MOV    R4,A
C:0x0AF8    22       RET
296: U16 uartNumbers(U16 toSend, bool transmit){           // Tx/Rx up to 4 length numbers over UART
297:     U16 out = toSend;
C:0x0AF9    8E08    MOV    0x08,R6
C:0x0AFB    8F09    MOV    0x09,R7
298:     U16 num = 0;
C:0xAFD    E4       CLR    A
C:0xAFE    FB       MOV    R3,A
C:0xAFF    FA       MOV    R2,A
299:     U16 scale = 10000;
300:     U8 test;
C:0xB00    902710  MOV    DPTR,#0x2710
301:     U8 i = 4;
C:0xB03    750B04  MOV    0x0B,#0x04
302:     bool bad = false;
C:0xB06    C201    CLR    0x20.1
303:     while(i){
// On zero done
C:0xB08    E50B    MOV    A,0x0B
C:0xB0A    6059    JZ     C:0B65
304:         scale /= 10;                                // Shift
C:0xB0C    7C00    MOV    R4,#0x00
C:0xB0E    7D0A    MOV    R5,#0x0A
C:0xB10    AF82    MOV    R7,0x82
C:0xB12    AE83    MOV    R6,0x83
C:0xB14    120A3E  LCALL  C?UIDIV(C:0A3E)
C:0xB17    8E83    MOV    0x83,R6
C:0xB19    8F82    MOV    0x82,R7
305:         i--;
C:0xB1B    150B    DEC    0x0B
306:         test = uart_in[i] - 48;                  // ascii to num
C:0xB1D    7432    MOV    A,#uart_in(0x32)
C:0xB1F    250B    ADD    A,0x0B
C:0xB21    F8       MOV    R0,A
C:0xB22    E6       MOV    A,@R0
C:0xB23    24D0    ADD    A,#0xD0
C:0xB25    F50A    MOV    0x0A,A
307:         if(test > 10){                           // check is 0 to 9, unsigned

```

```

C:0x0B27    D3      SETB    C
C:0x0B28    940A    SUBB    A,#0x0A
C:0x0B2A    4002    JC      C:0B2E
308:          bad = true;
C:0x0B2C    D201    SETB    0x20.1
309:          }
310:          num += test*scale;
// shift in to position
C:0x0B2E    AFOA    MOV     R7,0x0A
C:0x0B30    7E00    MOV     R6,#0x00
C:0x0B32    AD82    MOV     R5,0x82
C:0x0B34    AC83    MOV     R4,0x83
C:0x0B36    120A2C  LCALL   C?IMUL(C:0A2C)
C:0x0B39    EF      MOV     A,R7
C:0x0B3A    2B      ADD     A,R3
C:0x0B3B    FB      MOV     R3,A
C:0x0B3C    EE      MOV     A,R6
C:0x0B3D    3A      ADDC   A,R2
C:0x0B3E    FA      MOV     R2,A
311:          if(transmit){                                // Put if statement at back of loop to save on jumps
C:0x0B3F    3000C6  JNB    0x20.0,C:0B08
312:          num = out / scale;
// 10 powers
C:0x0B42    AE08    MOV     R6,0x08
C:0x0B44    AF09    MOV     R7,0x09
C:0x0B46    120A3E  LCALL   C?UIDIV(C:0A3E)
C:0x0B49    AA06    MOV     R2,0x06
C:0x0B4B    AB07    MOV     R3,0x07
313:          uartLoadOut(num + 48);                    // Number to ascii
C:0x0B4D    EF      MOV     A,R7
C:0x0B4E    2430    ADD     A,#adc3(0x30)
C:0x0B50    FF      MOV     R7,A
C:0x0B51    120B9C  LCALL   uartLoadOut(C:0B9C)
314:          out %= scale;                           // Remainder for next time
C:0x0B54    AE08    MOV     R6,0x08
C:0x0B56    AF09    MOV     R7,0x09
C:0x0B58    AD82    MOV     R5,0x82
C:0x0B5A    AC83    MOV     R4,0x83
C:0x0B5C    120A3E  LCALL   C?UIDIV(C:0A3E)
C:0x0B5F    8C08    MOV     0x08,R4
C:0x0B61    8D09    MOV     0x09,R5
315:          }
316:          }
C:0x0B63    80A3    SJMP   C:0B08
317:          uartLoadOut('\n');
C:0x0B65    7FOA    MOV     R7,#0x0A
C:0x0B67    120B9C  LCALL   uartLoadOut(C:0B9C)
318:          uartLoadOut('\r');
C:0x0B6A    7F0D    MOV     R7,#0x0D
C:0x0B6C    120B9C  LCALL   uartLoadOut(C:0B9C)
319:          if(bad){
C:0x0B6F    300104  JNB    0x20.1,C:0B76
320:          num = DEFAULT_OUT_MV;                      // Not all valid numbers so set output as default
C:0x0B72    7A13    MOV     R2,#0x13
C:0x0B74    7B88    MOV     R3,#TCON(0x88)
321:          }
322:          return num;
C:0x0B76    AE02    MOV     R6,0x02
C:0x0B78    AF03    MOV     R7,0x03
323:          }
324:
C:0x0B7A    22      RET
325: U16 readAdc(U8 sel){                                // Read the available ADCs
326:          U8 i;
327:          ADCOMX = sel;
C:0x0B7B    8FBB    MOV     ADCOMX(0xBB),R7
328:          for(i=0;i<2;i++){
C:0x0B7D    E4      CLR     A
C:0x0B7E    FF      MOV     R7,A
329:          ADCOCNO |= ADCOCNO_ADBUSY__SET;
C:0x0B7F    43E810  ORL     ADCOCNO(0xE8),#0x10
330:          while(ADCOCNO & ADCOCNO_ADBUSY__SET);        // Wait for sample to complete
C:0x0B82    E5E8    MOV     A,ADCOCNO(0xE8)
C:0x0B84    20E4FB  JB      0xE0.4,C:0B82
331:          }
C:0x0B87    OF      INC     R7
C:0x0B88    BF02F4  CJNE   R7,#0x02,C:0B7F
332:          return (((U32)ADCO)*SCALE_MUL) >> 10;           // Scale to mV
C:0x0B8B    AFBD    MOV     R7,ADCO(0xBD)
C:0x0B8D    AEBE    MOV     R6,0x8E
C:0x0B8F    7C17    MOV     R4,#0x17
C:0x0B91    7DB4    MOV     R5,#0xB4
C:0x0B93    120ACC  LCALL   C?LIMUL(C:0ACC)
C:0x0B96    780A    MOV     R0,#0x0A
C:0x0B98    120A93  LCALL   C?ULSHR(C:0A93)
333:          }
334:
335: -----
336: // Interrupt Routines
337: -----
338:
C:0x0B9B    22      RET
290: void uartLoadOut(U8 tx){
// Handle buffering out Tx UART
291:          uart_out[head] = tx;                            // Buffer outgoing
C:0x0B9C    AE23    MOV     R6,head(0x23)
C:0x0B9E    7439    MOV     A,#uart_out(0x39)
C:0x0BA0    2E      ADD     A,R6
C:0x0BA1    F8      MOV     R0,A

```

```

C :0x0BA2    A607      MOV      @R0 ,0x07
292:          head++;
C :0x0BA4    0523      INC      head(0x23)
293:          head %= UART_SIZE_OUT;           // Wrap around
C :0x0BA6    532307    ANL      head(0x23) ,#0x07
294: }
C :0x0BA9    22       RET
C :0x0BA9A   787F      MOV      R0 ,#0x7F
C :0x0BAC    E4       CLR      A
C :0x0BAD    F6       MOV      @R0 ,A
C :0x0BAE    D8FD      DJNZ    R0 ,C:OBAD
C :0x0BBO    758140    MOV      0x81 ,#0x40
C :0x0BB3    020800    LJMP   main(C:0800)
339: INTERRUPT (TIMER1_ISR, TIMER1_IRQn){}           // Needed for UART timing
C :0x0BB6    32       RETI

```

Listing 5: LIMUL

```

C?LIMUL:
C :0x0ACC    EC       MOV      A ,R4
C :0x0ACD    8EFO      MOV      0xF0 ,R6
C :0x0ACF    A4       MUL      AB
C :0x0ADO    CC       XCH      A ,R4
C :0x0AD1    C5F0      XCH      A ,0xF0
C :0x0AD3    CG       XCH      A ,R4
C :0x0AD4    CD       XCH      A ,R5
C :0x0AD5    F8       MOV      R0 ,A
C :0x0AD6    EF       MOV      A ,R7
C :0x0AD7    A4       MUL      AB
C :0x0AD8    CE       XCH      A ,R6
C :0x0AD9    C5F0      XCH      A ,0xF0
C :0x0ADB    2D       ADD      A ,R5
C :0x0ADC    FD       MOV      R5 ,A
C :0x0ADD    E4       CLR      A
C :0x0ADE    3C       ADDC     A ,R4
C :0x0ADF    FC       MOV      R4 ,A
C :0x0AE0    E8       MOV      A ,R0
C :0x0AE1    A4       MUL      AB
C :0x0AE2    2E       ADD      A ,R6
C :0x0AE3    C8       XCH      A ,R0
C :0x0AE4    C5F0      XCH      A ,0xF0
C :0x0AE6    3D       ADDC     A ,R5
C :0x0AE7    FD       MOV      R5 ,A
C :0x0AE8    E4       CLR      A
C :0x0AE9    3C       ADDC     A ,R4
C :0x0AEA    FC       MOV      R4 ,A
C :0x0AEB    EF       MOV      A ,R7
C :0x0AEC    A4       MUL      AB
C :0x0AED    FF       MOV      R7 ,A
C :0x0AEE    E5F0      MOV      A ,0xF0
C :0x0AF0    28       ADD      A ,R0
C :0x0AF1    FE       MOV      R6 ,A
C :0x0AF2    E4       CLR      A
C :0x0AF3    3D       ADDC     A ,R5
C :0x0AF4    FD       MOV      R5 ,A
C :0x0AF5    E4       CLR      A
C :0x0AF6    3C       ADDC     A ,R4
C :0x0AF7    FC       MOV      R4 ,A
C :0x0AF8    22       RET

```

Listing 6: ULSHR

```

C?ULSHR:
C :0x0A93    E8       MOV      A ,R0
C :0x0A94    600F      JZ      C:0AA5
C :0x0A96    EC       MOV      A ,R4
C :0x0A97    C3       CLR      C
C :0x0A98    13       RRC      A
C :0x0A99    FC       MOV      R4 ,A
C :0x0A9A    ED       MOV      A ,R5
C :0x0A9B    13       RRC      A
C :0x0A9C    FD       MOV      R5 ,A
C :0x0A9D    EE       MOV      A ,R6
C :0x0A9E    13       RRC      A
C :0x0A9F    FE       MOV      R6 ,A
C :0x0AA0    EF       MOV      A ,R7
C :0x0AA1    13       RRC      A
C :0x0AA2    FF       MOV      R7 ,A
C :0x0AA3    D8F1      DJNZ    R0 ,C:0A96
C :0x0AA5    22       RET

```

Listing 7: IMUL

```

C?IMUL:
C :0x0A2C    EF       MOV      A ,R7
C :0x0A2D    8DFO      MOV      0xF0 ,R5
C :0x0A2F    A4       MUL      AB
C :0x0A30    A8FO      MOV      R0 ,0xF0
C :0x0A32    CF       XCH      A ,R7
C :0x0A33    8CF0      MOV      0xF0 ,R4
C :0x0A35    A4       MUL      AB
C :0x0A36    28       ADD      A ,R0
C :0x0A37    CE       XCH      A ,R6

```

C :0x0A38	8DFO	MOV	0xF0 ,R5
C :0x0A3A	A4	MUL	AB
C :0x0A3B	2E	ADD	A ,R6
C :0x0A3C	FE	MOV	R6 ,A
C :0x0A3D	22	RET	

Listing 8: UDIV

C?UIDIV:			
C :0x0A3E	BC000B	CJNE	R4 ,#0x00 ,C:0A4C
C :0x0A41	BE0029	CJNE	R6 ,#0x00 ,C:0A6D
C :0x0A44	EF	MOV	A ,R7
C :0x0A45	8DFO	MOV	0xF0 ,R5
C :0x0A47	84	DIV	AB
C :0x0A48	FF	MOV	R7 ,A
C :0x0A49	ADFO	MOV	R5 ,0xF0
C :0x0A4B	22	RET	
C :0x0A4C	E4	CLR	A
C :0x0A4D	CC	XCH	A ,R4
C :0x0A4E	F8	MOV	R0 ,A
C :0x0A4F	75F008	MOV	0xF0 ,#0x08
C :0x0A52	EF	MOV	A ,R7
C :0x0A53	2F	ADD	A ,R7
C :0x0A54	FF	MOV	R7 ,A
C :0x0A55	EE	MOV	A ,R6
C :0x0A56	33	RLC	A
C :0x0A57	FE	MOV	R6 ,A
C :0x0A58	EC	MOV	A ,R4
C :0x0A59	33	RLC	A
C :0x0A5A	FC	MOV	R4 ,A
C :0x0A5B	EE	MOV	A ,R6
C :0x0A5C	9D	SUBB	A ,R5
C :0x0A5D	EC	MOV	A ,R4
C :0x0A5E	98	SUBB	A ,R0
C :0x0A5F	4005	JC	C:0A66
C :0x0A61	FC	MOV	R4 ,A
C :0x0A62	EE	MOV	A ,R6
C :0x0A63	9D	SUBB	A ,R5
C :0x0A64	FE	MOV	R6 ,A
C :0x0A65	OF	INC	R7
C :0x0A66	D5F0E9	DJNZ	0xF0 ,C:0A52
C :0x0A69	E4	CLR	A
C :0x0A6A	CE	XCH	A ,R6
C :0x0A6B	FD	MOV	R5 ,A
C :0x0A6C	22	RET	
C :0x0A6D	ED	MOV	A ,R5
C :0x0A6E	F8	MOV	R0 ,A
C :0x0A6F	F5F0	MOV	0xF0 ,A
C :0x0A71	EE	MOV	A ,R6
C :0x0A72	84	DIV	AB
C :0x0A73	20D21C	JB	0xD0 .2 ,C:0A92
C :0x0A76	FE	MOV	R6 ,A
C :0x0A77	ADFO	MOV	R5 ,0xF0
C :0x0A79	75F008	MOV	0xF0 ,#0x08
C :0x0A7C	EF	MOV	A ,R7
C :0x0A7D	2F	ADD	A ,R7
C :0x0A7E	FF	MOV	R7 ,A
C :0x0A7F	ED	MOV	A ,R5
C :0x0A80	33	RLC	A
C :0x0A81	FD	MOV	R5 ,A
C :0x0A82	4007	JC	C:0A8B
C :0x0A84	98	SUBB	A ,R0
C :0x0A85	5006	JNC	C:0A8D
C :0x0A87	D5FOF2	DJNZ	0xF0 ,C:0A7C
C :0x0A8A	22	RET	
C :0x0A8B	C3	CLR	C
C :0x0A8C	98	SUBB	A ,R0
C :0x0A8D	FD	MOV	R5 ,A
C :0x0A8E	OF	INC	R7
C :0x0A8F	D5FOEA	DJNZ	0xF0 ,C:0A7C
C :0x0A92	22	RET	

Listing 9: CCASE

C?CCASE:			
C :0x0AA6	D083	POP	0x83
C :0x0AA8	D082	POP	0x82
C :0x0AAA	F8	MOV	R0 ,A
C :0x0AAB	E4	CLR	A
C :0x0AAC	93	MOVC	A ,@A+DPTR
C :0x0AAD	7012	JNZ	C:0AC1
C :0x0AAF	7401	MOV	A ,#0x01
C :0x0AB1	93	MOVC	A ,@A+DPTR
C :0x0AB2	700D	JNZ	C:0AC1
C :0x0AB4	A3	INC	DPTR
C :0x0AB5	A3	INC	DPTR
C :0x0AB6	93	MOVC	A ,@A+DPTR
C :0x0AB7	F8	MOV	R0 ,A
C :0x0AB8	7401	MOV	A ,#0x01
C :0x0ABA	93	MOVC	A ,@A+DPTR
C :0x0ABB	F582	MOV	0x82 ,A
C :0x0ABD	8883	MOV	0x83 ,R0
C :0x0ABF	E4	CLR	A
C :0x0AC0	73	JMP	@A+DPTR
C :0x0AC1	7402	MOV	A ,#0x02

C :0x0AC3	93	MOVC	A , @A+DPTR
C :0x0AC4	68	XRL	A ,R0
C :0x0AC5	60EF	JZ	C :0AB6
C :0x0AC7	A3	INC	DPTR
C :0x0AC8	A3	INC	DPTR
C :0x0AC9	A3	INC	DPTR
C :0x0ACA	80DF	SJMP	C :0AAB
