21BDS0340

Abhinav Dinesh Srivatsa

Microprocessors and Microcontrollers Lab

Task – V

**Question 1**

Aim:

Write a program that continuously gets 8-bit data from P0 and sends it to P1 while simultaneously creating a square wave of 0.5 ms period on pin P2.1. Use Interrupt and enable Timer 1 to create the square wave.

Tools Required:

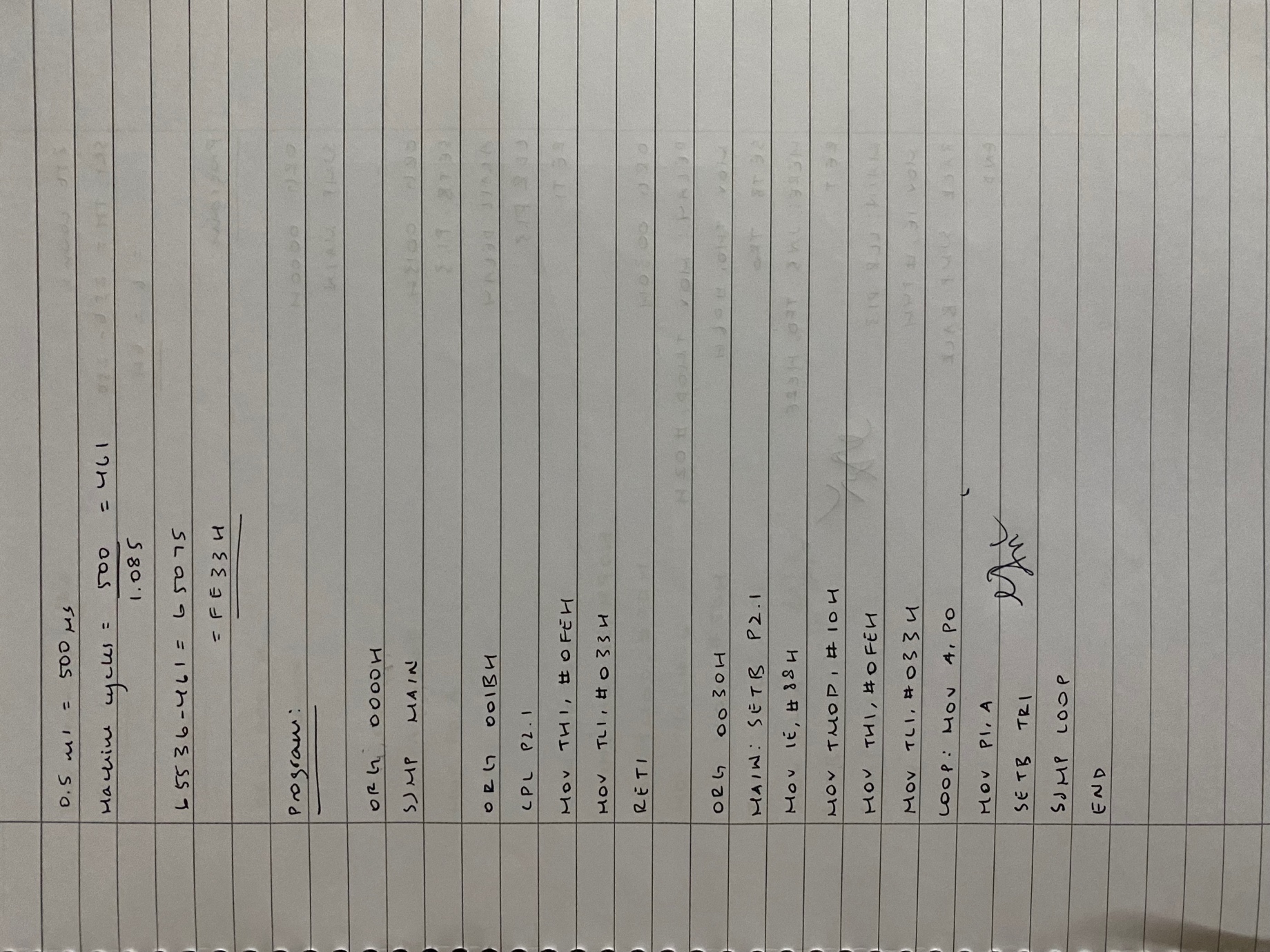
8051 microcontroller

Keil microcontroller software

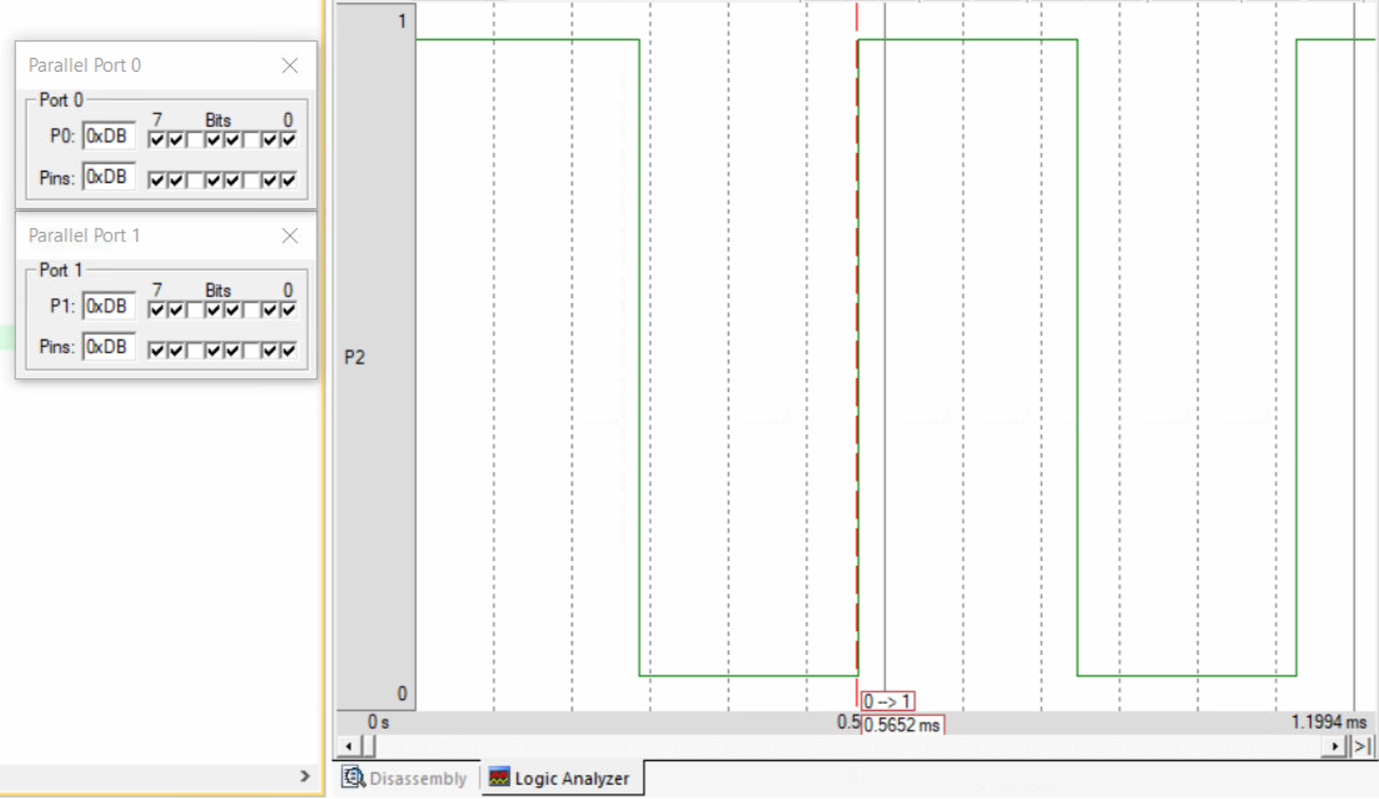
Program

|  |  |  |  |
| --- | --- | --- | --- |
| Memory Locations | Label | Mnemonics | Comments |
|  |  | ORG 0000H |  |
| 0000H |  | LJMP MAIN | Jump to MAIN |
|  |  | ORG 001BH | Address called if interrupt T1 occurred |
| 001BH |  | CPL P2.1 | Complement P2.1 |
| 001DH |  | MOV TH1, #0FEH | Move values for timer |
| 0020H |  | MOV TL1, #33H |  |
| 0023H |  | RETI | Return and clear interrupt flags |
|  |  | ORG 0030H |  |
| 0030H | MAIN: | SETB P2.1 | Set bit P2.1 to high |
| 0032H |  | MOV IE, #88H | Set interrupt as enabled for T1 |
| 0035H |  | MOV TMOD, #10H | Set TMOD for timer 1, mode 1 |
| 0038H |  | MOV TH1, #0FEH | Move values for timer |
| 003BH |  | MOV TL1, #33H |  |
| 003EH | LOOP: | MOV A, P0 | Move value from P0 to A |
| 0040H |  | MOV P1, A | Move value from A to P1 |
| 0042H |  | SETB TR1 | Start timer 1 |
| 0044H |  | SJMP LOOP | Jump to LOOP |
|  |  | END |  |

Manual Calculations:



Output:



Result:

This program generates a 0.5 ms square wave on port 2.1 and transfers values between port 0 to port 1.

**Question 2**

Aim:

Assume that the INT1 pin is connected to a switch that is normally high. Whenever it goes low, it should turn on an LED. The LED is connected to P1.3 and is normally off. When it is turned on it should stay on for a 250 count. Use External Interrupt 1.

Tools Required:

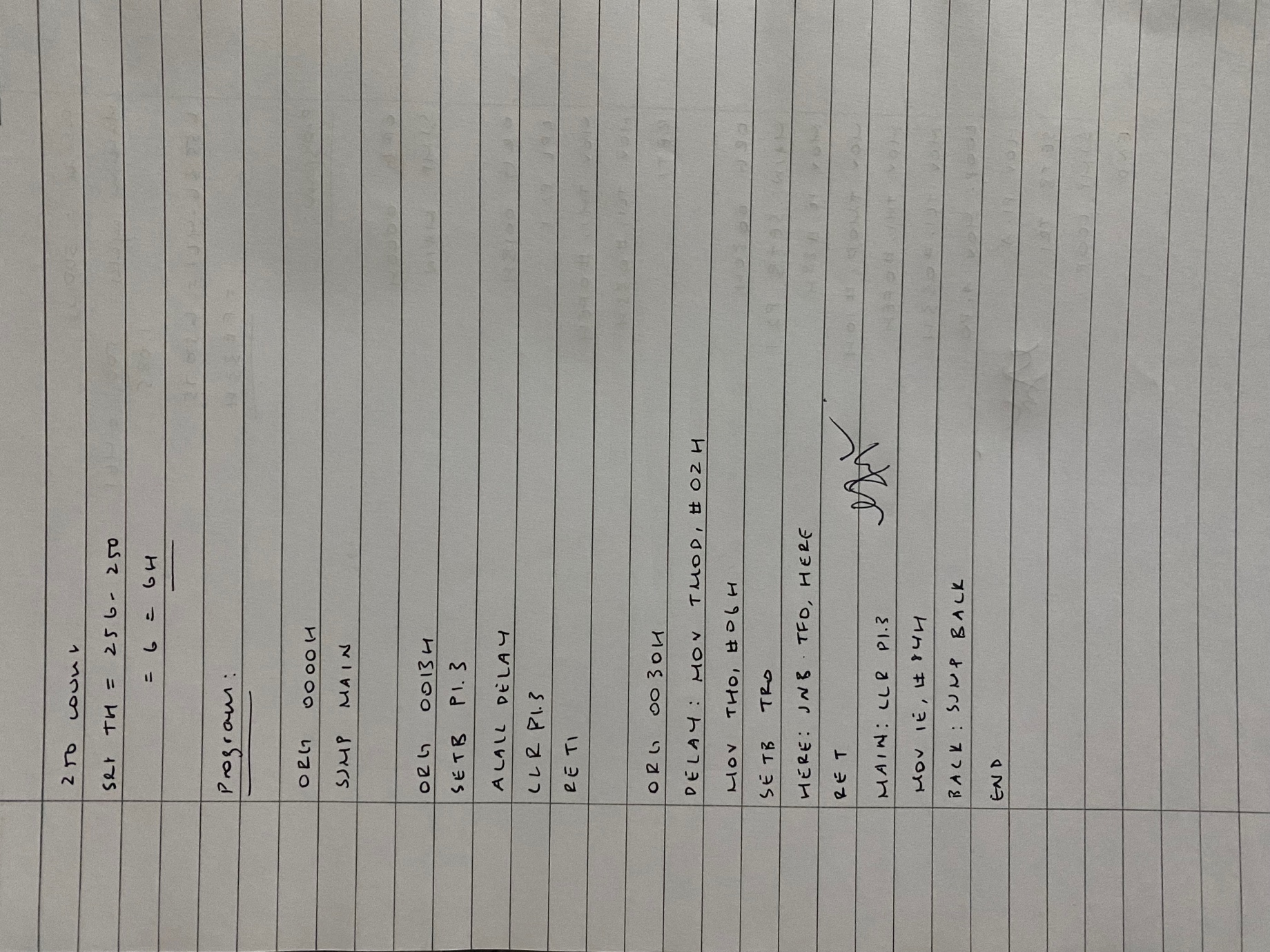
8051 microcontroller

Keil microcontroller software

Program

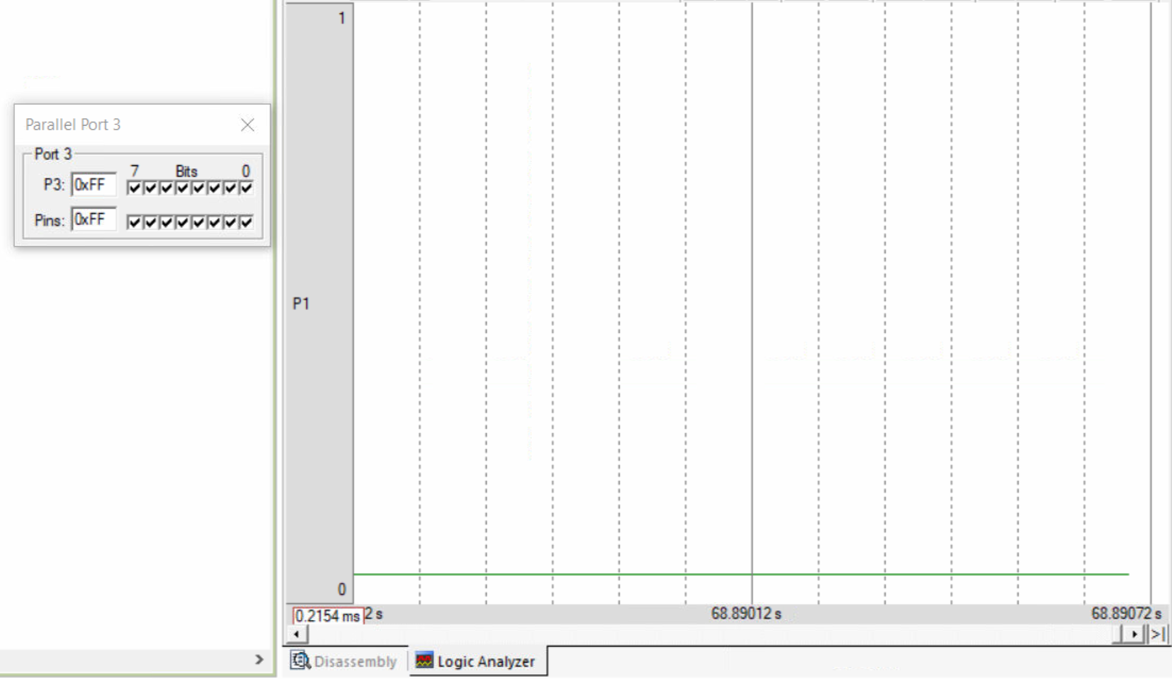
|  |  |  |  |
| --- | --- | --- | --- |
| Memory Locations | Label | Mnemonics | Comments |
|  |  | ORG 0000H |  |
| 0000H |  | SJMP MAIN | Jump to MAIN |
|  |  | ORG 0013H | Address called if interrupt IT1 occurred |
| 0013H |  | SETB P1.3 | Set P1.3 to high |
| 0015H |  | ACALL DELAY | Call DELAY |
| 0017H |  | CLR P1.3 | Set P1.3 to low |
| 0019H |  | RETI | Return and clear interrupt flags |
|  |  | ORG 0030H |  |
| 0030H | DELAY: | MOV TMOD, #02H | Set TMOD to timer 0 mode 2 |
| 0033H |  | MOV TH0, #06H | Set values for timer 0 |
| 0036H |  | SETB TR0 |  |
| 0038H | HERE: | JNB TF0, HERE | Loop here until TF0 high |
| 003BH |  | RET | Return |
| 003CH | MAIN: | CLR P1.3 | Set P1.3 to low |
| 003EH |  | MOV IE, #84H | Set interrupts enabled for IT1 (external) |
| 0041H | BACK: | SJMP BACK | Loop here forever |
|  |  | END |  |

Manual Calculations:

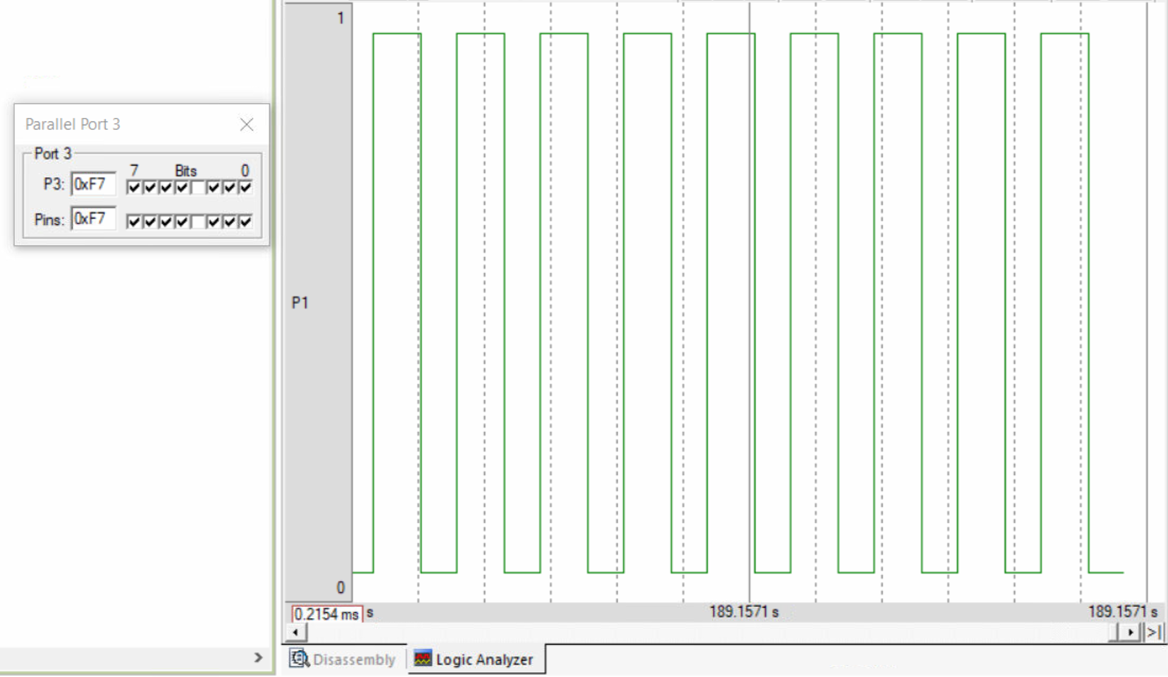


Output:

When P3.3 high:



When P3.3 low:



Result:

This program generates a square wave of 250 count when the external interrupt is low, otherwise does nothing.

**Question 3**

Aim:

Write a program in which the 8051 gets data from P1 and sends it to P2 continuously while incoming data from the serial port is sent to P0. Enable serial interrupt and set the baud rate at 4800.

Tools Required:

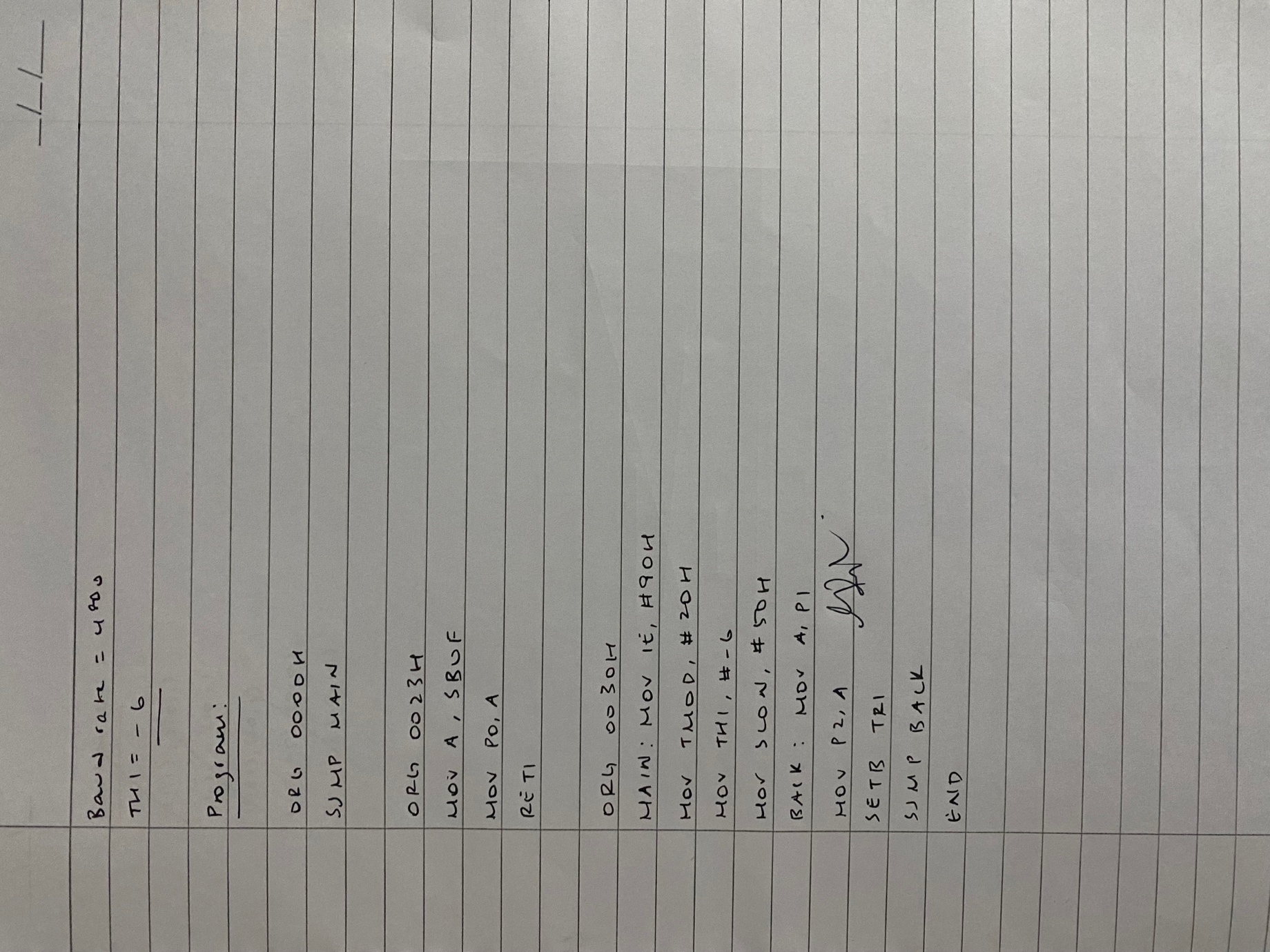
8051 microcontroller

Keil microcontroller software

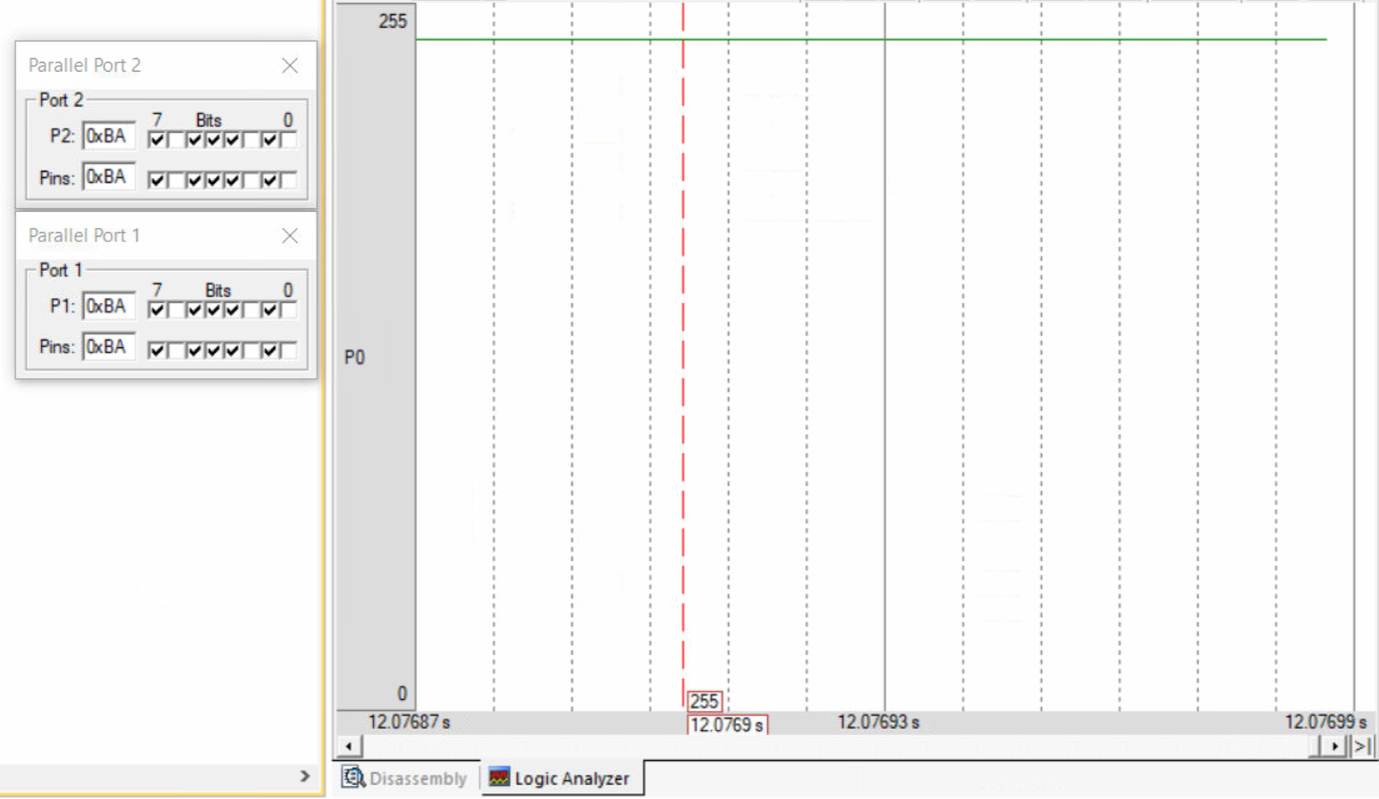
Program

|  |  |  |  |
| --- | --- | --- | --- |
| Memory Locations | Label | Mnemonics | Comments |
|  |  | ORG 0000H |  |
| 0000H |  | SJMP MAIN | Jump to MAIN |
|  |  | ORG 0023H | Address called if interrupt TI/RI occurred |
| 0023H |  | MOV A, SBUF | Move serial data to A |
| 0025H |  | MOV P0, A | Move A to P0 |
| 0027H |  | RETI | Return and clear interrupt flags |
|  |  | ORG 0030H |  |
| 0030H | MAIN: | MOV IE, #90H | Set interrupts enabled and TI/RI |
| 0033H |  | MOV TMOD, #20H | Set TMOD for timer 1, mode 2 |
| 0036H |  | MOV TH1, #-6 | Set baud rate to 4800 |
| 0039H |  | MOV SCON, #50H | Set SCON for serial communication |
| 003CH | BACK: | MOV A, P1 | Move P1 to A |
| 003EH |  | MOV P2, A | Move A to P2 |
| 0040H |  | SETB TR1 | Start timer 1 |
| 0042H |  | SJMP BACK | Jump to BACK |
|  |  | END |  |

Manual Calculations:



Output:



Result:

This program transfers data from P1 to P2 and transfers serial data to P0.