dhanwanth07 / Square-Wave-Generation-Using-8051

Code Pull requests Actions Projects Wiki Security Insights

main Square-Wave-Generation-Using-8051 / README.md

dhanwanth07 Update README.md c77c7fd · now

55 lines (49 loc) · 2.08 KB

Preview Code Blame Raw

Square Wave Generation Using 8051

Aim:

To generate a square wave using the 8051 microcontroller and observe the waveform using Keil software and Proteus simulation.

Apparatus Required:

Laptop with Keil uVision software
Proteus Design Suite

Circuit Diagram Setup in Proteus:

1. Open Proteus and create a new project.
2. Add the following components from the library:
   8051 Microcontroller (AT89C51)
   Oscilloscope (to observe the waveform)
   Resistor (1kΩ) (if using hardware)
3. Connect P1.0 of the microcontroller to the oscilloscope's input.
4. Save the design and proceed to programming in Keil.

Algorithm:

1. Configure P1.0 as an output port.
2. Set P1.0 HIGH to generate a HIGH pulse.
3. Introduce a delay.
4. Set P1.0 LOW to generate a LOW pulse.
5. Introduce a delay.
6. Repeat the process continuously.

Program:

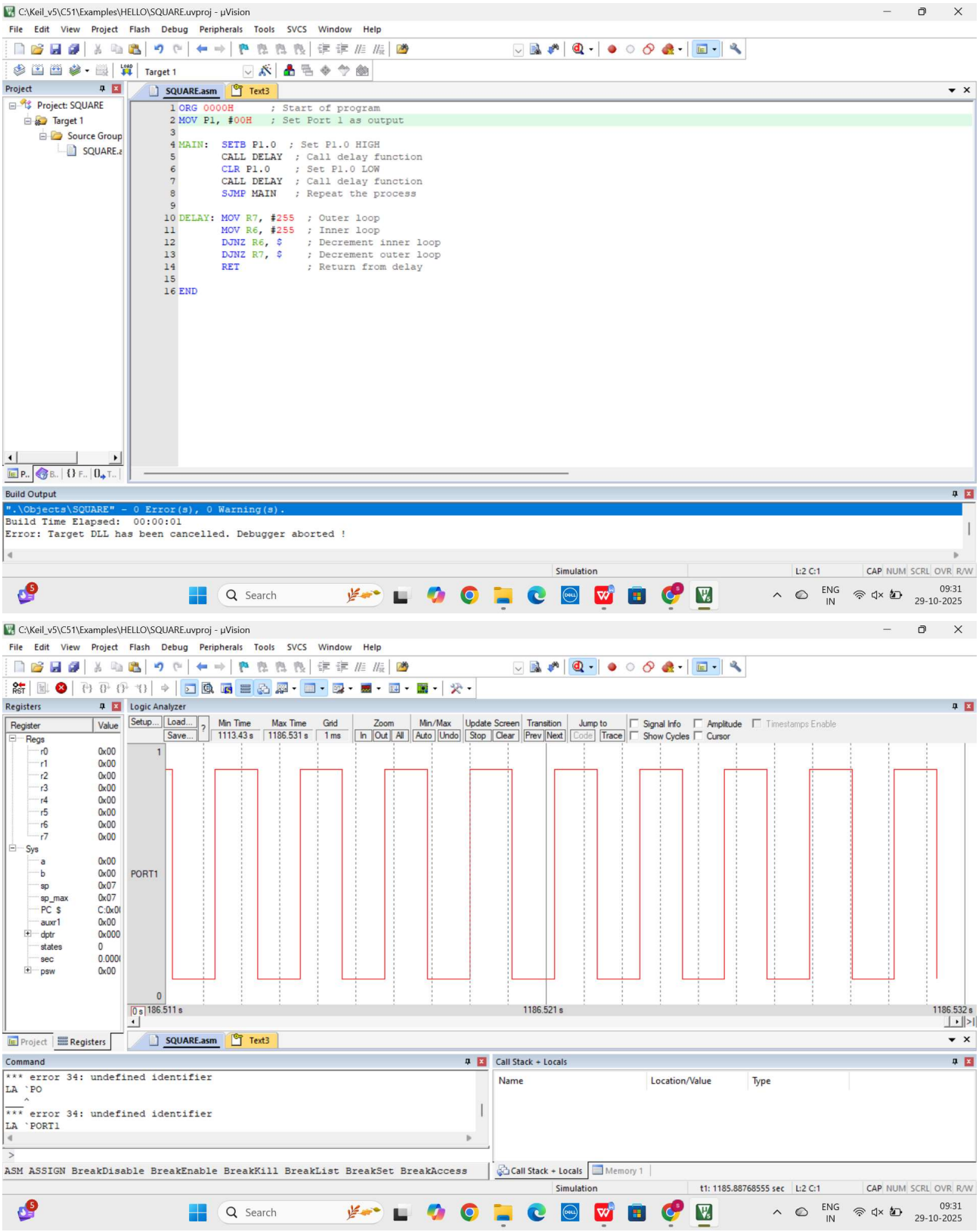
```
Program (Keil - 8051 Assembly)
ORG 0000H      ; Start of program
MOV P1, #00H   ; Set Port 1 as output

MAIN:  SETB P1.0  ; Set P1.0 HIGH
       CALL DELAY  ; Call delay function
       CLR P1.0    ; Set P1.0 LOW
       CALL DELAY  ; Call delay function
       SJMP MAIN   ; Repeat the process

DELAY: MOV R7, #255  ; Outer loop
       MOV R6, #255  ; Inner loop
       DJNZ R6, $    ; Decrement inner loop
       DJNZ R7, $    ; Decrement outer loop
       RET           ; Return from delay

END
```

Simulation in Proteus:

1. Open Proteus and load the HEX file generated from Keil.
2. Connect P1.0 to an oscilloscope.
3. Run the simulation and observe the square wave on the oscilloscope.
4. Adjust delay loops if needed to modify the wave frequency.

Output:

A square wave will be observed on the oscilloscope with a defined time period.



Result:

The square wave generation using the 8051 microcontroller has been successfully implemented and simulated using Keil and Proteus.