FWD – Advanced Embedded Systems Course

Project #2 – RTOS

❖ **Rubric Requirement4 : “Verify the system Implementation”**

➢ **System Hyperperiod Calculation :**

- Task1 Periodicity = 50
- Task2 Periodicity = 50
- Task3 Periodicity = 100
- Task4 Periodicity = 20
- Task5 Periodicity = 10
- Task6 Periodicity = 100

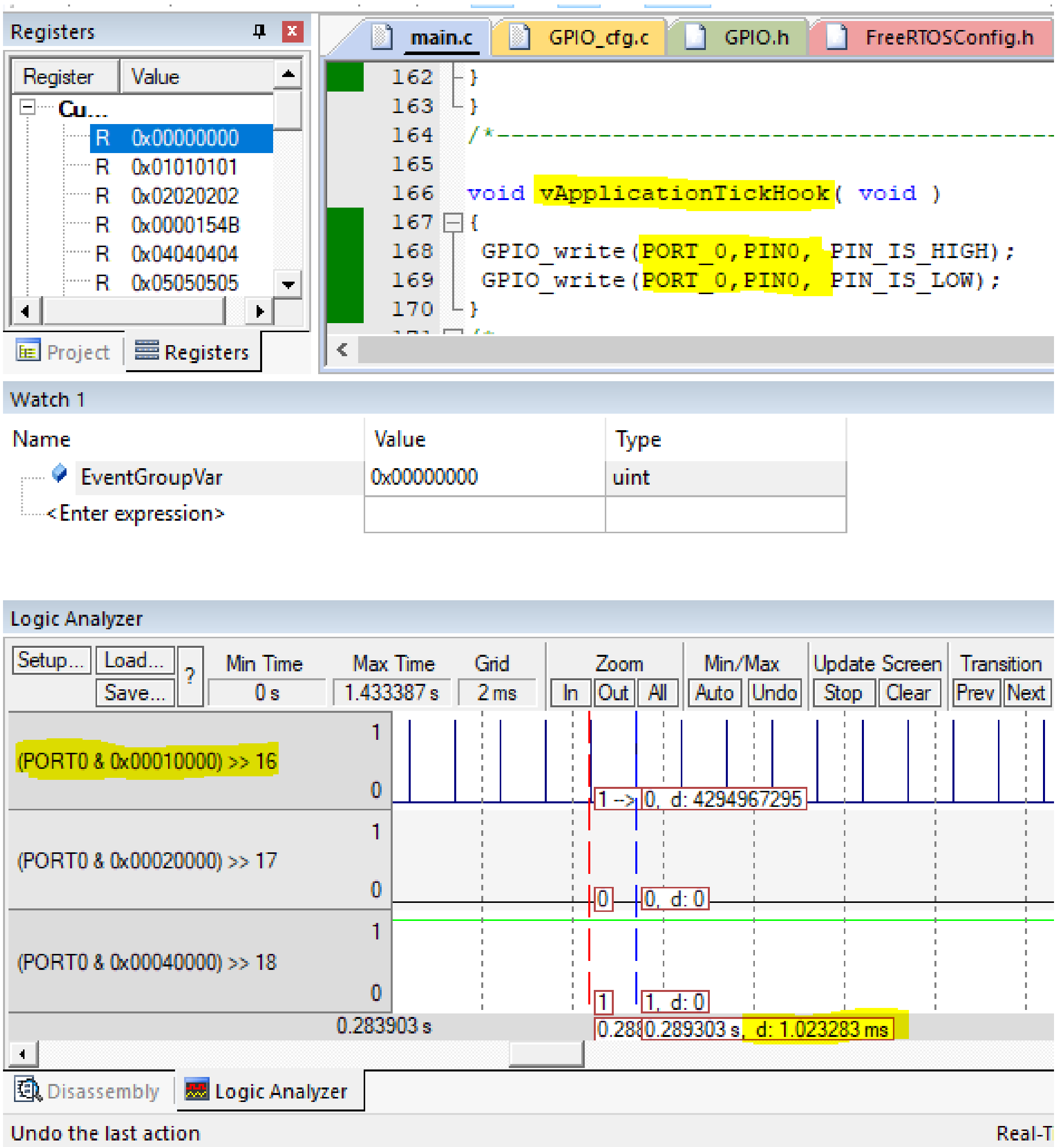
⇨ Hyperperiod = 100ms , The least significant multiplier of all tasks periodicity

➢ System CPU Load :

Tasks practical Periodicity and Execution time:

- Tick:

> Tick period = 1ms



Register	Value
Cu...	
R	0x00000000
R	0x01010101
R	0x02020202
R	0x0000154B
R	0x04040404
R	0x05050505

```
}
}
/*-----------------------------------------

void vApplicationTickHook( void )
{
  GPIO_write(PORT_0,PIN0, PIN_IS_HIGH);
  GPIO_write(PORT_0,PIN0, PIN_IS_LOW);
}
```

Watch 1

Name	Value	Type
EventGroupVar	0x00000000	uint
<Enter expression>		

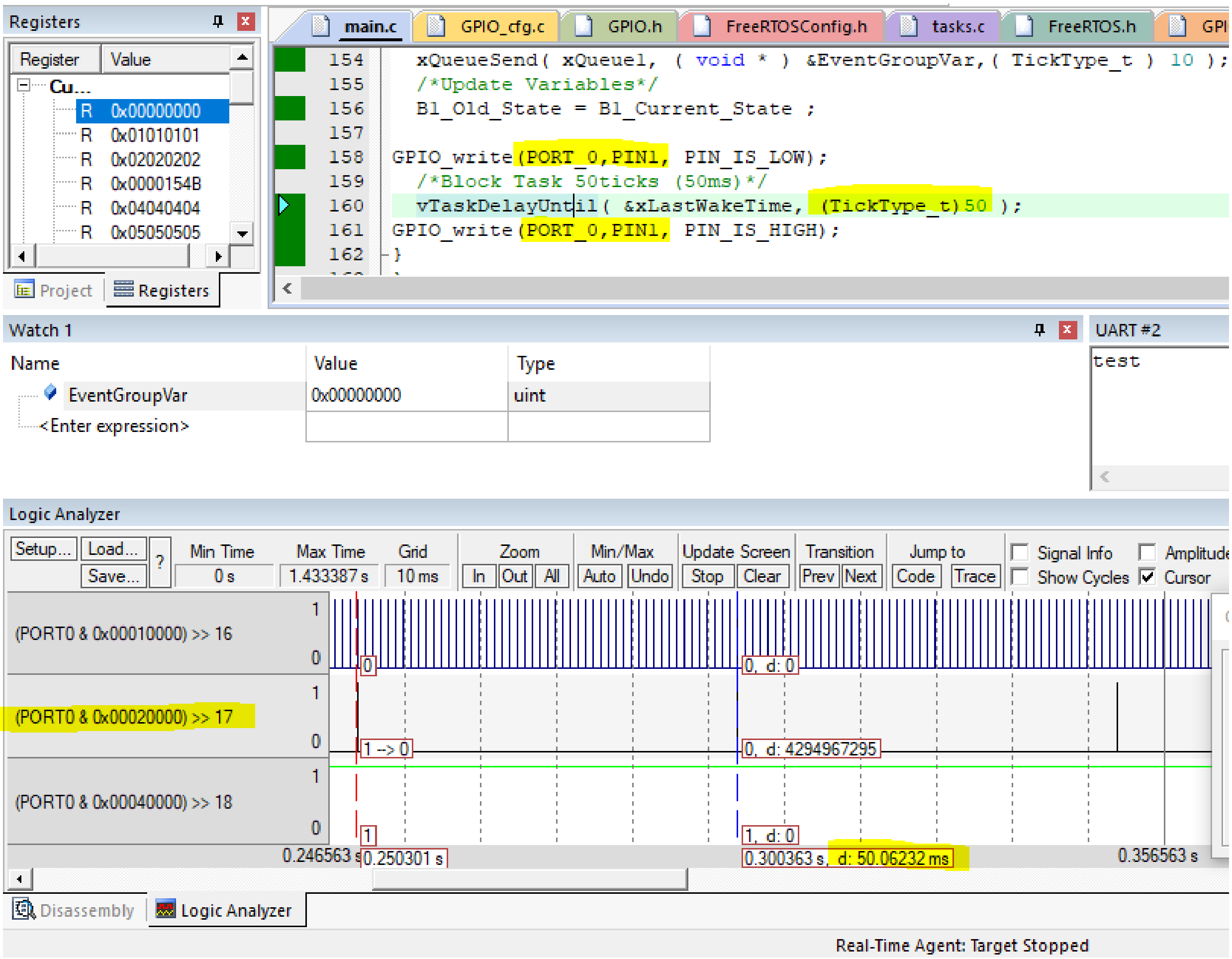
Min Time: 0 s
Max Time: 1.433387 s
Grid: 2 ms

(PORT0 & 0x00010000) >> 16 — 1 / 0 — 1 ->| 0, d: 4294967295
(PORT0 & 0x00020000) >> 17 — 1 / 0 — 0 | 0, d: 0
(PORT0 & 0x00040000) >> 18 — 1 / 0 — 1 | 1, d: 0

0.283903 s | 0.288 | 0.289303 s, d: 1.023283 ms

- Task 1

→ Periodicity = 50 ms



```
    xQueueSend( xQueue1, ( void * ) &EventGroupVar,( TickType_t ) 10 );
    /*Update Variables*/
    B1_Old_State = B1_Current_State ;

GPIO_write(PORT_0,PIN1, PIN_IS_LOW);
    /*Block Task 50ticks (50ms)*/
    vTaskDelayUntil( &xLastWakeTime, (TickType_t)50 );
GPIO_write(PORT_0,PIN1, PIN_IS_HIGH);
}
```

Register	Value
Cu...	
R	0x00000000
R	0x01010101
R	0x02020202
R	0x0000154B
R	0x04040404
R	0x05050505

Watch 1

Name	Value	Type
EventGroupVar	0x00000000	uint
<Enter expression>		

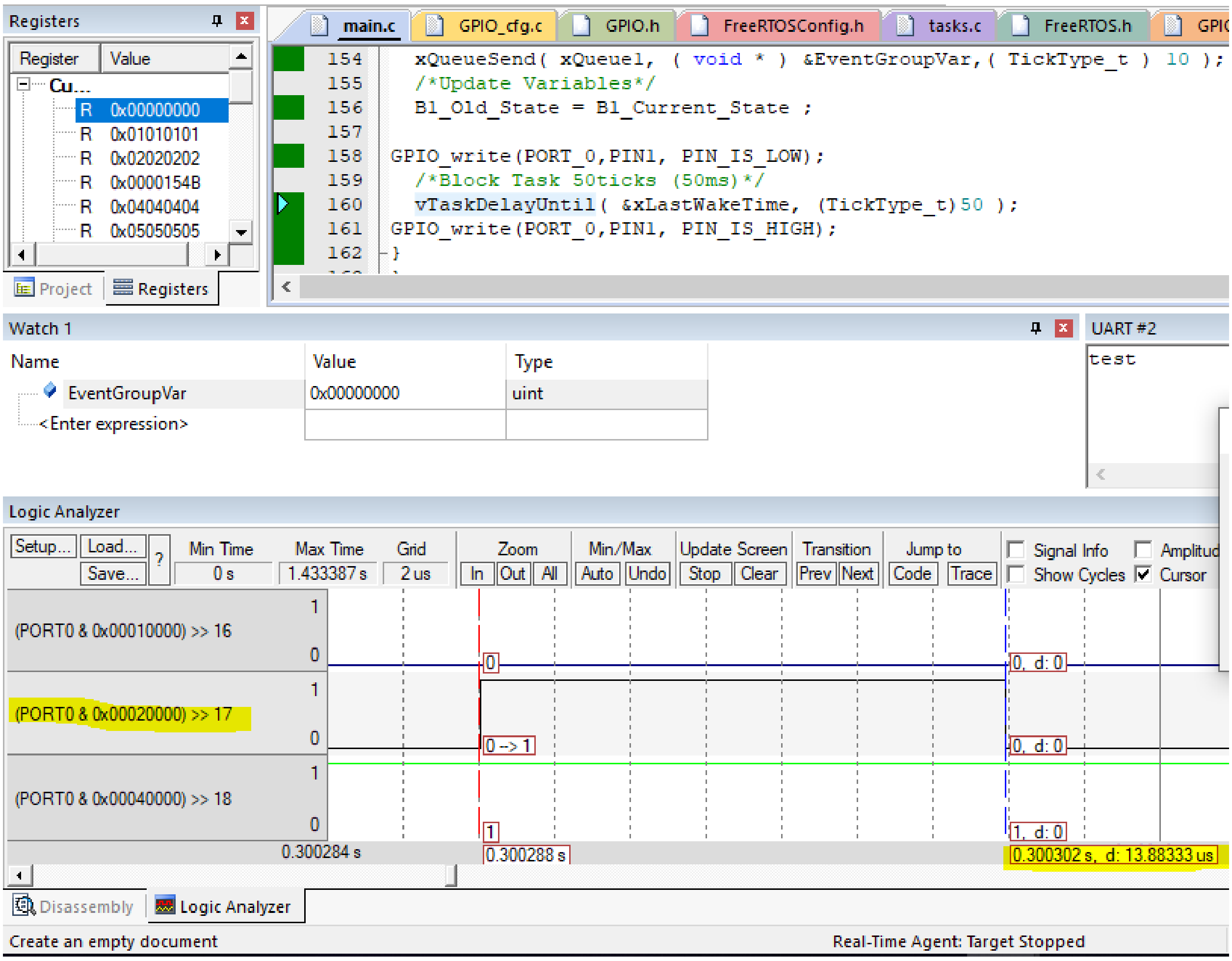
UART #2

```
test
```

Logic Analyzer

Min Time	Max Time	Grid
0 s	1.433387 s	10 ms

→ Execution Time =13.9 us



```
    xQueueSend( xQueue1, ( void * ) &EventGroupVar,( TickType_t ) 10 );
    /*Update Variables*/
    B1_Old_State = B1_Current_State ;

GPIO_write(PORT_0,PIN1, PIN_IS_LOW);
    /*Block Task 50ticks (50ms)*/
    vTaskDelayUntil( &xLastWakeTime, (TickType_t)50 );
GPIO_write(PORT_0,PIN1, PIN_IS_HIGH);
}
```

Register	Value
Cu...	
R	0x00000000
R	0x01010101
R	0x02020202
R	0x0000154B
R	0x04040404
R	0x05050505

Watch 1

Name	Value	Type
EventGroupVar	0x00000000	uint
<Enter expression>		

UART #2

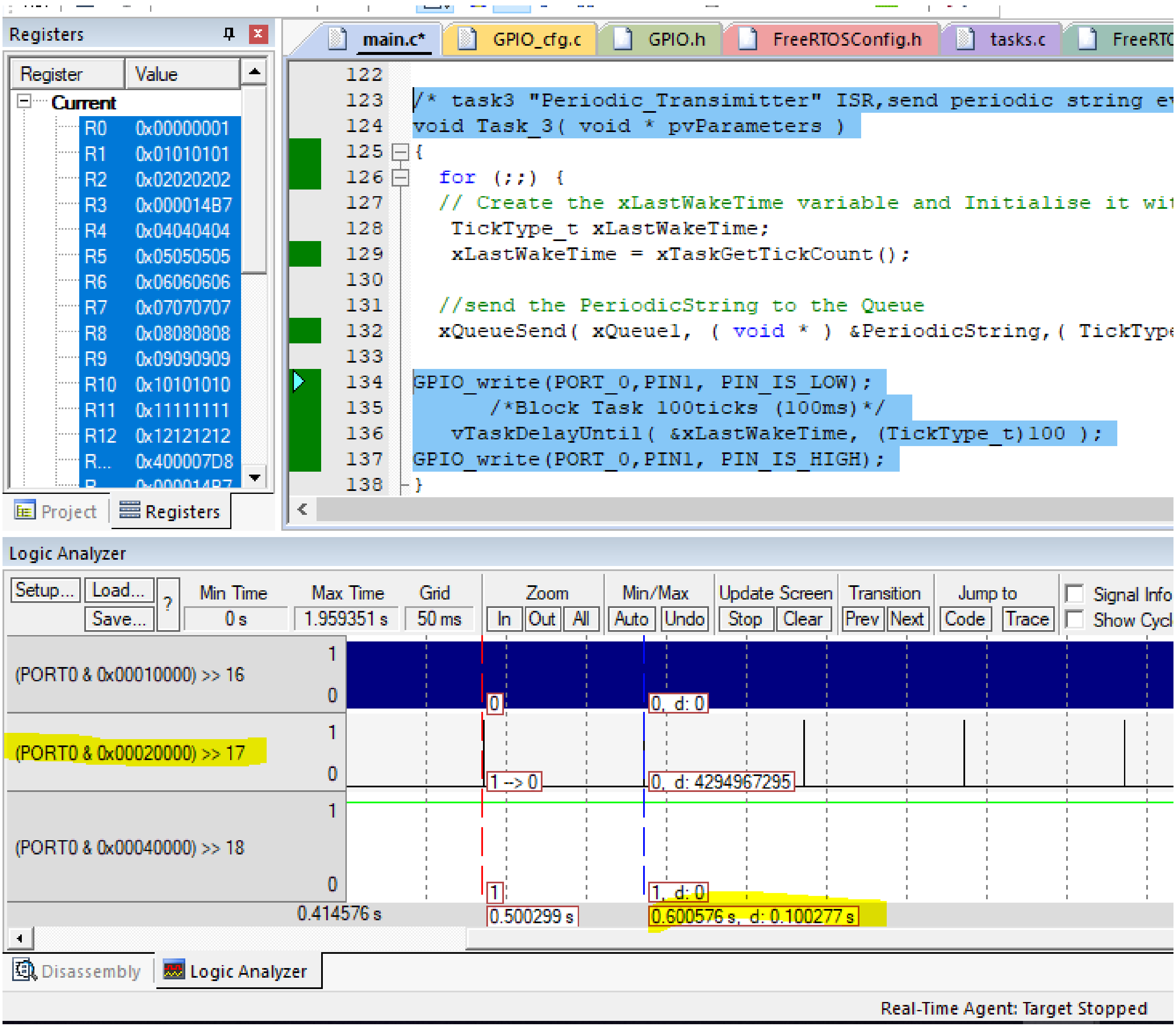
```
test
```

Logic Analyzer

Min Time	Max Time	Grid
0 s	1.433387 s	2 us

- Task2 : Same as Task1
  → Periodicity = 50 ms
  → Execution Time =13.9 us

- Task3 :
  → Periodicity = 100ms (0.1s)



Register	Value
Current	
R0	0x00000001
R1	0x01010101
R2	0x02020202
R3	0x000014B7
R4	0x04040404
R5	0x05050505
R6	0x06060606
R7	0x07070707
R8	0x08080808
R9	0x09090909
R10	0x10101010
R11	0x11111111
R12	0x12121212
R...	0x400007D8
R...	0x000014B7

```
/* task3 "Periodic_Transimitter" ISR,send periodic string e
void Task_3( void * pvParameters )
{
  for (;;) {
  // Create the xLastWakeTime variable and Initialise it wi
   TickType_t xLastWakeTime;
   xLastWakeTime = xTaskGetTickCount();

  //send the PeriodicString to the Queue
  xQueueSend( xQueue1, ( void * ) &PeriodicString,( TickType

GPIO_write(PORT_0,PIN1, PIN_IS_LOW);
      /*Block Task 100ticks (100ms)*/
   vTaskDelayUntil( &xLastWakeTime, (TickType_t)100 );
GPIO_write(PORT_0,PIN1, PIN_IS_HIGH);
}
```

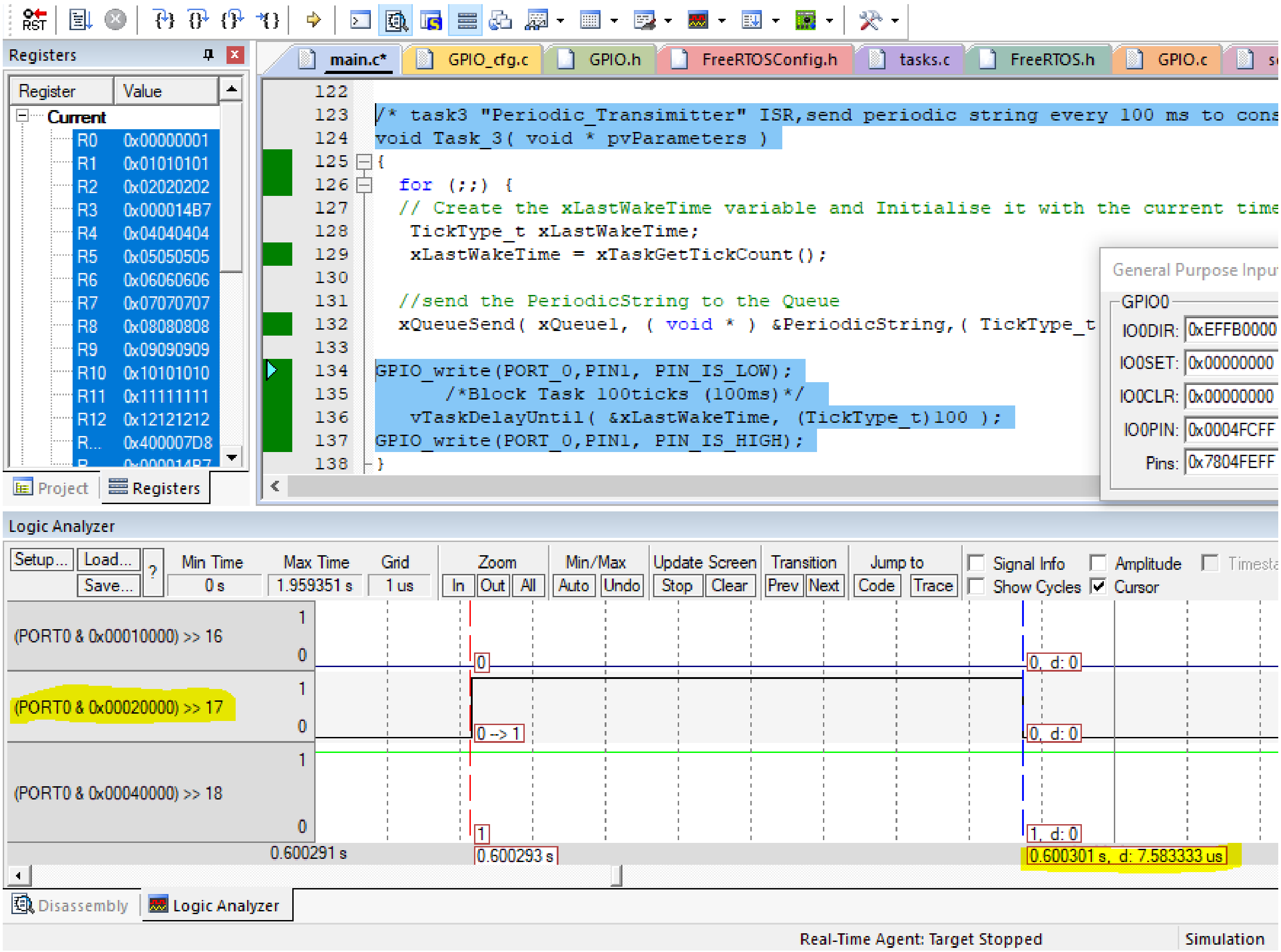
Min Time 0 s
Max Time 1.959351 s
Grid 50 ms
□ Signal Info
□ Show Cycl

(PORT0 & 0x00010000) >> 16 1 0
(PORT0 & 0x00020000) >> 17 1 0
(PORT0 & 0x00040000) >> 18 1 0

0 0, d: 0
1 -> 0 0, d: 4294967295
1 1, d: 0
0.414576 s 0.500299 s 0.600576 s, d: 0.100277 s

Real-Time Agent: Target Stopped

→ Execution Time = 7.5 us



Register	Value
Current	
R0	0x00000001
R1	0x01010101
R2	0x02020202
R3	0x000014B7
R4	0x04040404
R5	0x05050505
R6	0x06060606
R7	0x07070707
R8	0x08080808
R9	0x09090909
R10	0x10101010
R11	0x11111111
R12	0x12121212
R...	0x400007D8

```
/* task3 "Periodic_Transimitter" ISR,send periodic string every 100 ms to cons
void Task_3( void * pvParameters )
{
  for (;;) {
  // Create the xLastWakeTime variable and Initialise it with the current time
   TickType_t xLastWakeTime;
   xLastWakeTime = xTaskGetTickCount();

  //send the PeriodicString to the Queue
  xQueueSend( xQueue1, ( void * ) &PeriodicString,( TickType_t

GPIO_write(PORT_0,PIN1, PIN_IS_LOW);
      /*Block Task 100ticks (100ms)*/
   vTaskDelayUntil( &xLastWakeTime, (TickType_t)100 );
GPIO_write(PORT_0,PIN1, PIN_IS_HIGH);
}
```

General Purpose Inpu

GPIO0

IO0DIR:	0xEFFB0000
IO0SET:	0x00000000
IO0CLR:	0x00000000
IO0PIN:	0x0004FCFF
Pins:	0x7804FEFF

Min Time	Max Time	Grid
0 s	1.959351 s	1 us

(PORT0 & 0x00010000) >> 16

(PORT0 & 0x00020000) >> 17

(PORT0 & 0x00040000) >> 18

0.600291 s

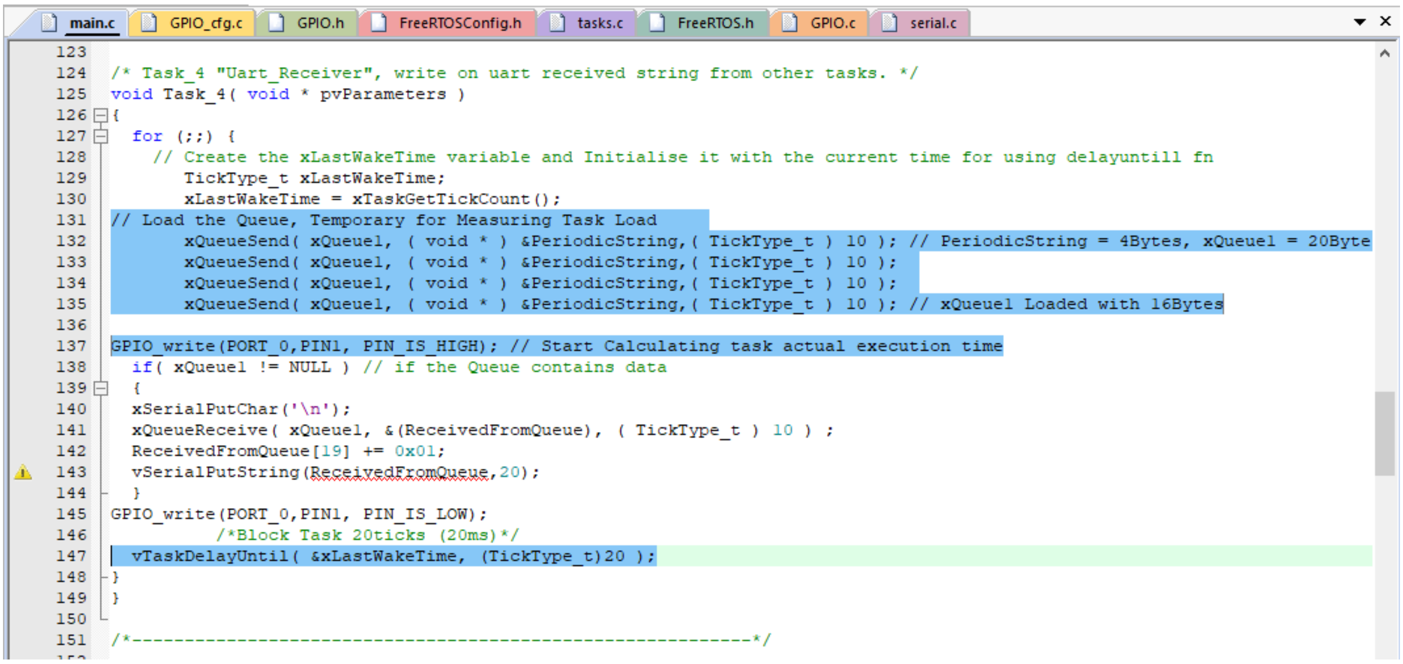
0.600293 s

0.600301 s, d: 7.583333 us

Real-Time Agent: Target Stopped

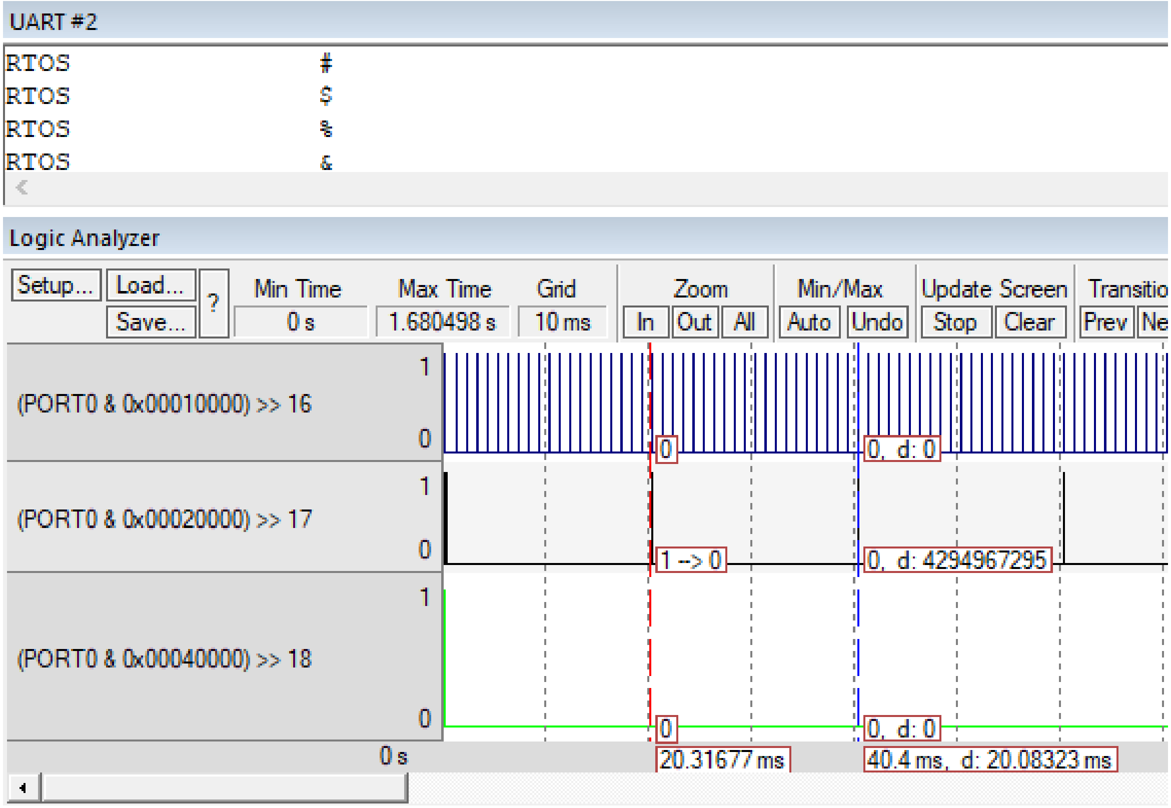
Simulation

- Task4 :
  → As we need the Queue to be full in order to test the Task actual load, i will Load the Queue every time inside the task body first, then start monitoring task with GPIOs starting from its particular job code only



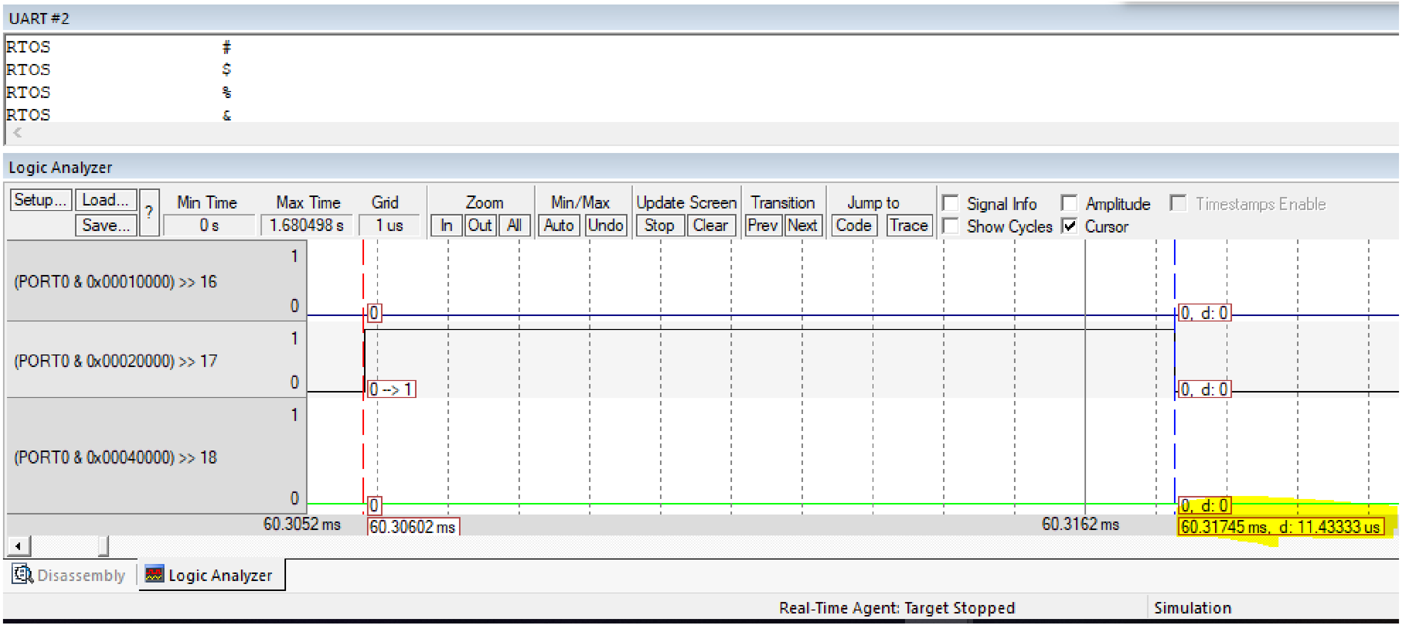
```
/* Task_4 "Uart_Receiver", write on uart received string from other tasks. */
void Task_4( void * pvParameters )
{
  for (;;) {
    // Create the xLastWakeTime variable and Initialise it with the current time for using delayuntill fn
       TickType_t xLastWakeTime;
       xLastWakeTime = xTaskGetTickCount();
// Load the Queue, Temporary for Measuring Task Load
       xQueueSend( xQueue1, ( void * ) &PeriodicString,( TickType_t ) 10 ); // PeriodicString = 4Bytes, xQueue1 = 20Byte
       xQueueSend( xQueue1, ( void * ) &PeriodicString,( TickType_t ) 10 );
       xQueueSend( xQueue1, ( void * ) &PeriodicString,( TickType_t ) 10 );
       xQueueSend( xQueue1, ( void * ) &PeriodicString,( TickType_t ) 10 ); // xQueue1 Loaded with 16Bytes

GPIO_write(PORT_0,PIN1, PIN_IS_HIGH); // Start Calculating task actual execution time
  if( xQueue1 != NULL ) // if the Queue contains data
  {
  xSerialPutChar('\n');
  xQueueReceive( xQueue1, &(ReceivedFromQueue), ( TickType_t ) 10 ) ;
  ReceivedFromQueue[19] += 0x01;
  vSerialPutString(ReceivedFromQueue,20);
  }
GPIO_write(PORT_0,PIN1, PIN_IS_LOW);
          /*Block Task 20ticks (20ms)*/
  vTaskDelayUntil( &xLastWakeTime, (TickType_t)20 );
 }
}

/*-------------------------------------------------------------*/
```

→ Periodicity = 20ms



```
RTOS              #
RTOS              $
RTOS              %
RTOS              &
```

→ Periodic time =~ 11.5us



UART #2

```
RTOS                #
RTOS                $
RTOS                %
RTOS                &
```

Logic Analyzer

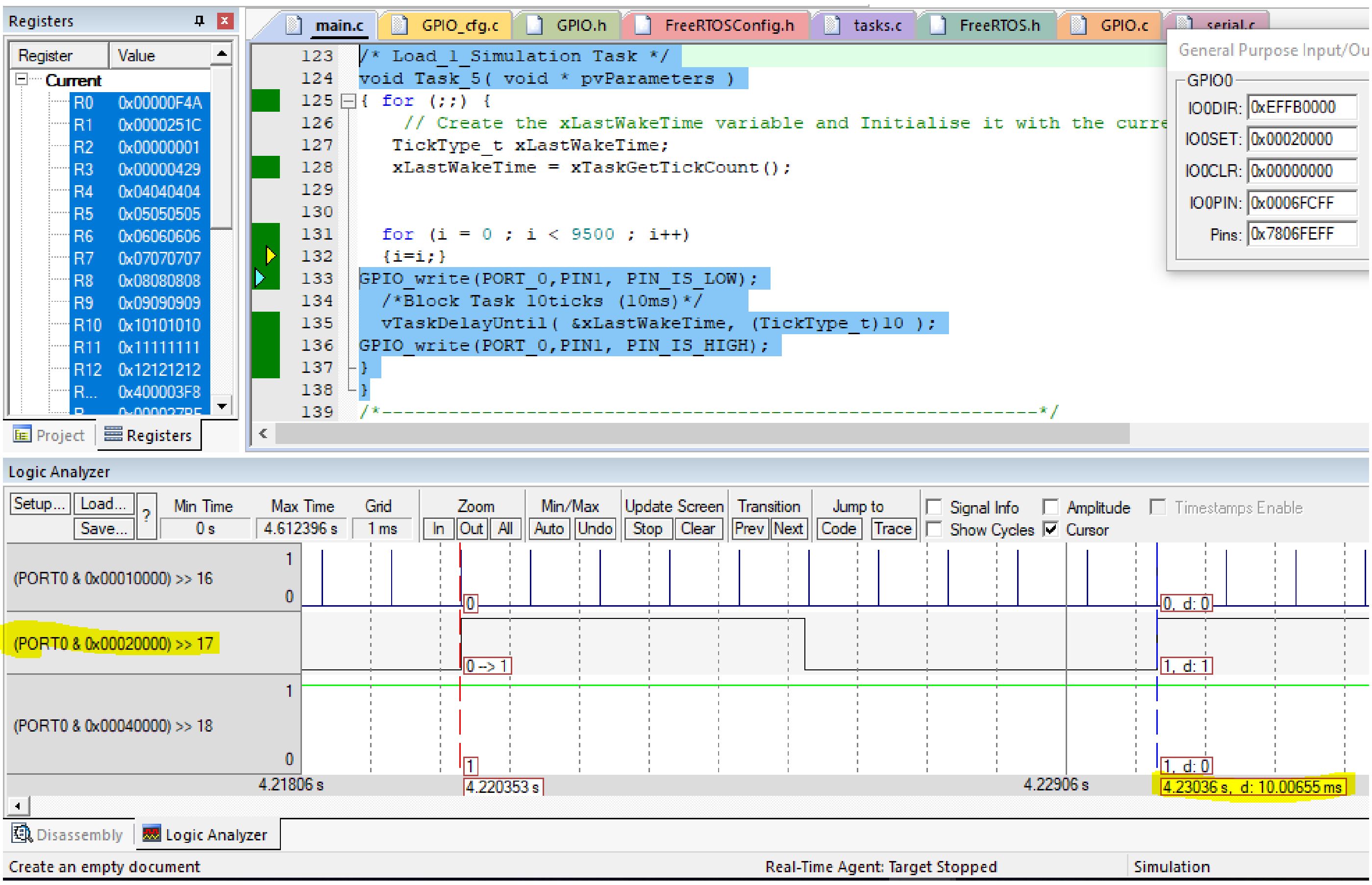
Min Time: 0 s; Max Time: 1.680498 s; Grid: 1 us

□ Signal Info □ Amplitude □ Timestamps Enable
□ Show Cycles ☑ Cursor

Real-Time Agent: Target Stopped Simulation

- Task5 :

→ Periodicity = 10ms



Registers

Register	Value
Current	
R0	0x00000F4A
R1	0x0000251C
R2	0x00000001
R3	0x00000429
R4	0x04040404
R5	0x05050505
R6	0x06060606
R7	0x07070707
R8	0x08080808
R9	0x09090909
R10	0x10101010
R11	0x11111111
R12	0x12121212
R...	0x400003F8

```
/* Load_1_Simulation Task */
void Task_5( void * pvParameters )
{ for (;;) {
    // Create the xLastWakeTime variable and Initialise it with the curre
    TickType_t xLastWakeTime;
    xLastWakeTime = xTaskGetTickCount();


  for (i = 0 ; i < 9500 ; i++)
  {i=i;}
GPIO_write(PORT_0,PIN1, PIN_IS_LOW);
  /*Block Task 10ticks (10ms)*/
  vTaskDelayUntil( &xLastWakeTime, (TickType_t)10 );
GPIO_write(PORT_0,PIN1, PIN_IS_HIGH);
}
}
/*-----------------------------------------------------------*/
```

General Purpose Input/Ou

GPIO0

IO0DIR:	0xEFFB0000
IO0SET:	0x00020000
IO0CLR:	0x00000000
IO0PIN:	0x0006FCFF
Pins:	0x7806FEFF

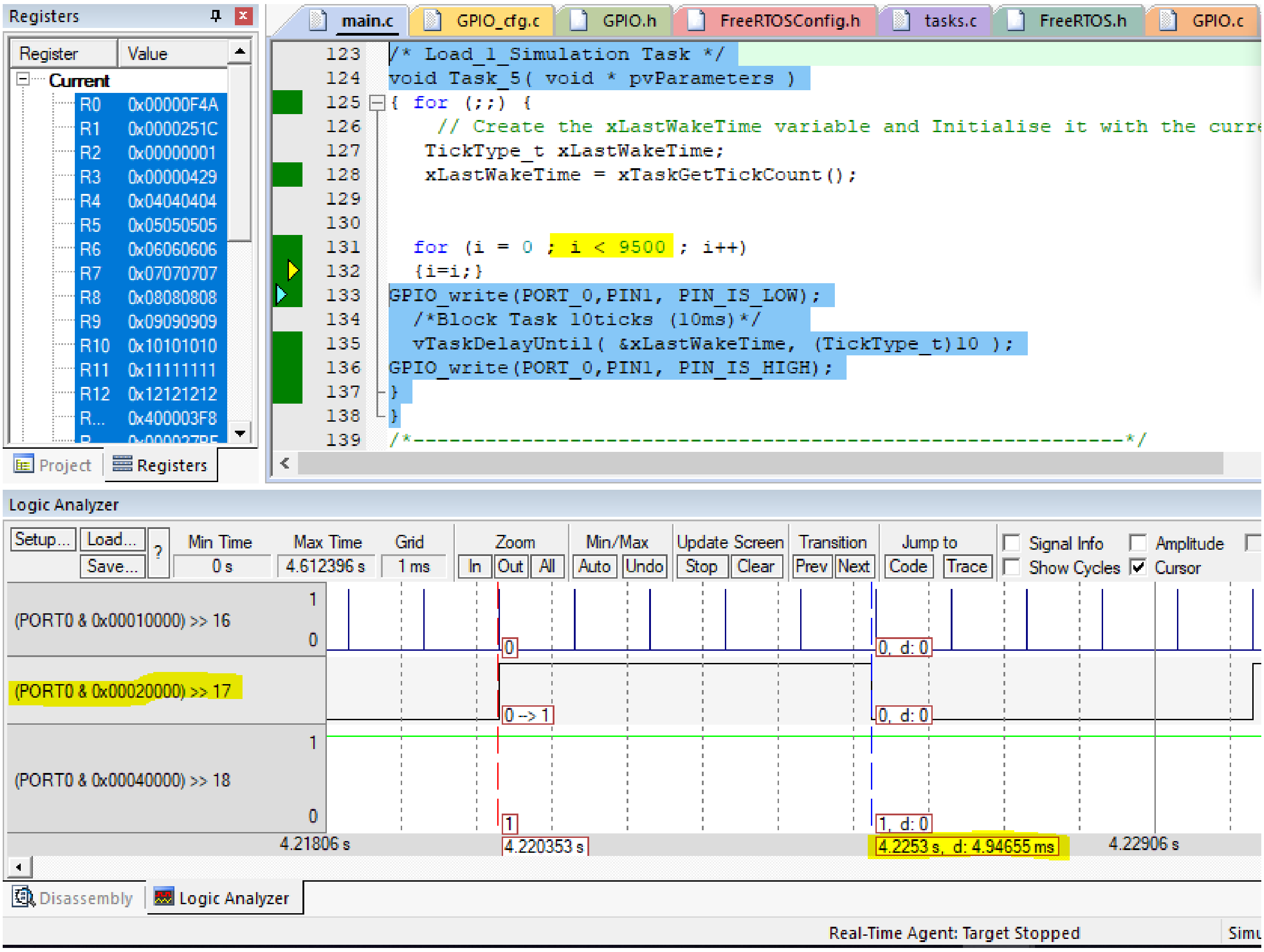
Logic Analyzer

Min Time: 0 s; Max Time: 4.612396 s; Grid: 1 ms

□ Signal Info □ Amplitude □ Timestamps Enable
□ Show Cycles ☑ Cursor

Create an empty document Real-Time Agent: Target Stopped Simulation

→ Execution =~ 5ms



Registers

Register	Value
Current	
R0	0x00000F4A
R1	0x0000251C
R2	0x00000001
R3	0x00000429
R4	0x04040404
R5	0x05050505
R6	0x06060606
R7	0x07070707
R8	0x08080808
R9	0x09090909
R10	0x10101010
R11	0x11111111
R12	0x12121212
R...	0x400003F8

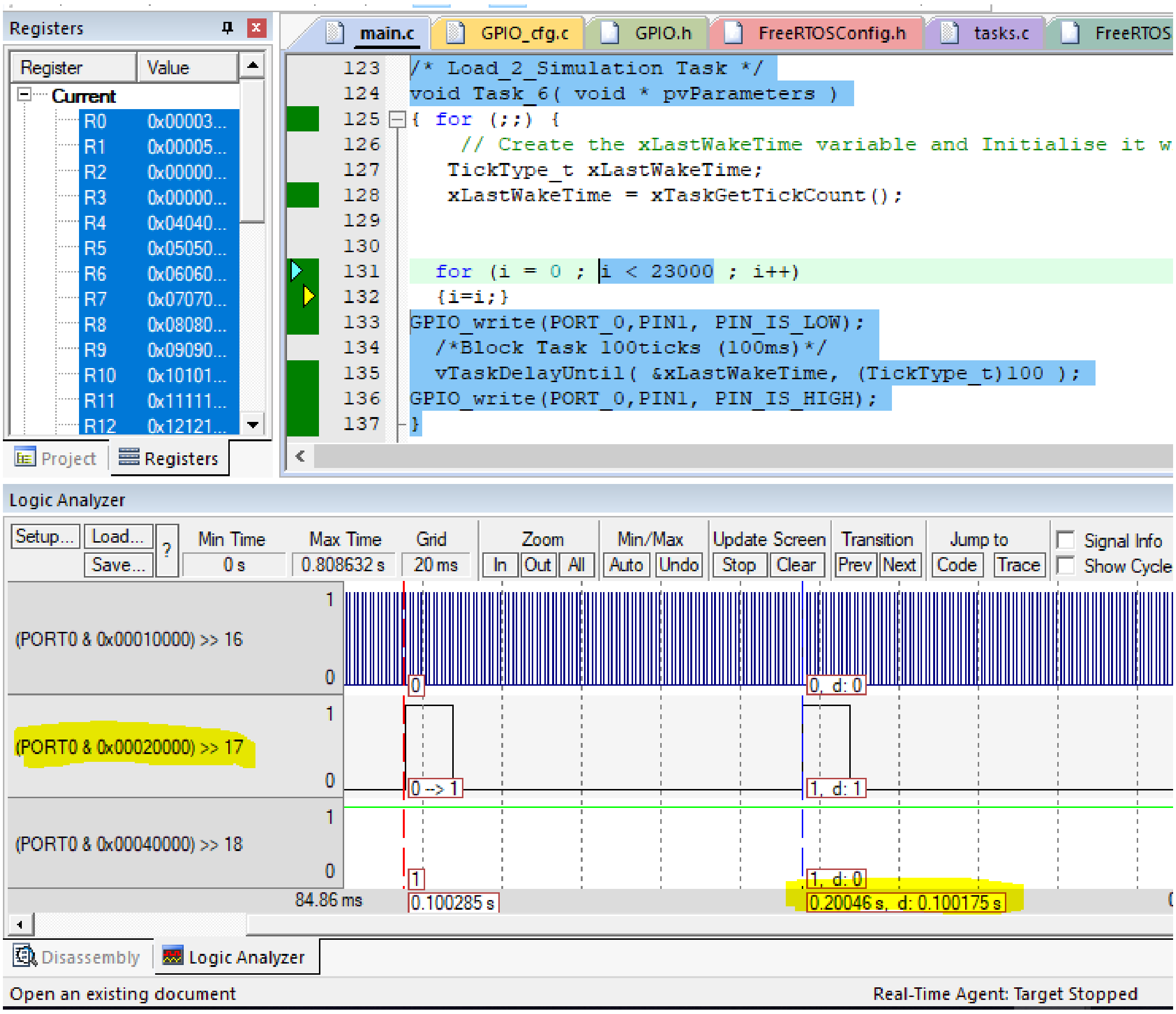
```
/* Load_1_Simulation Task */
void Task_5( void * pvParameters )
{ for (;;) {
    // Create the xLastWakeTime variable and Initialise it with the curre
   TickType_t xLastWakeTime;
   xLastWakeTime = xTaskGetTickCount();


  for (i = 0 ; i < 9500 ; i++)
  {i=i;}
GPIO_write(PORT_0,PIN1, PIN_IS_LOW);
  /*Block Task 10ticks (10ms)*/
  vTaskDelayUntil( &xLastWakeTime, (TickType_t)10 );
GPIO_write(PORT_0,PIN1, PIN_IS_HIGH);
}
}
/*-----------------------------------------------------------*/
```

Min Time: 0 s
Max Time: 4.612396 s
Grid: 1 ms
☐ Signal Info
☐ Show Cycles
☐ Amplitude
☑ Cursor

Real-Time Agent: Target Stopped

- Task6 :

→ Periodicity = 100ms



Register	Value
Current	
R0	0x00003...
R1	0x00005...
R2	0x00000...
R3	0x00000...
R4	0x04040...
R5	0x05050...
R6	0x06060...
R7	0x07070...
R8	0x08080...
R9	0x09090...
R10	0x10101...
R11	0x11111...
R12	0x12121...

```c
/* Load 2 Simulation Task */
void Task_6( void * pvParameters )
{ for (;;) {
   // Create the xLastWakeTime variable and Initialise it w
  TickType_t xLastWakeTime;
  xLastWakeTime = xTaskGetTickCount();


  for (i = 0 ; i < 23000 ; i++)
  {i=i;}
GPIO_write(PORT_0,PIN1, PIN_IS_LOW);
  /*Block Task 100ticks (100ms)*/
  vTaskDelayUntil( &xLastWakeTime, (TickType_t)100 );
GPIO_write(PORT_0,PIN1, PIN_IS_HIGH);
}
```

Min Time 0 s | Max Time 0.808632 s | Grid 20 ms

(PORT0 & 0x00010000) >> 16 — 0 | 0, d: 0

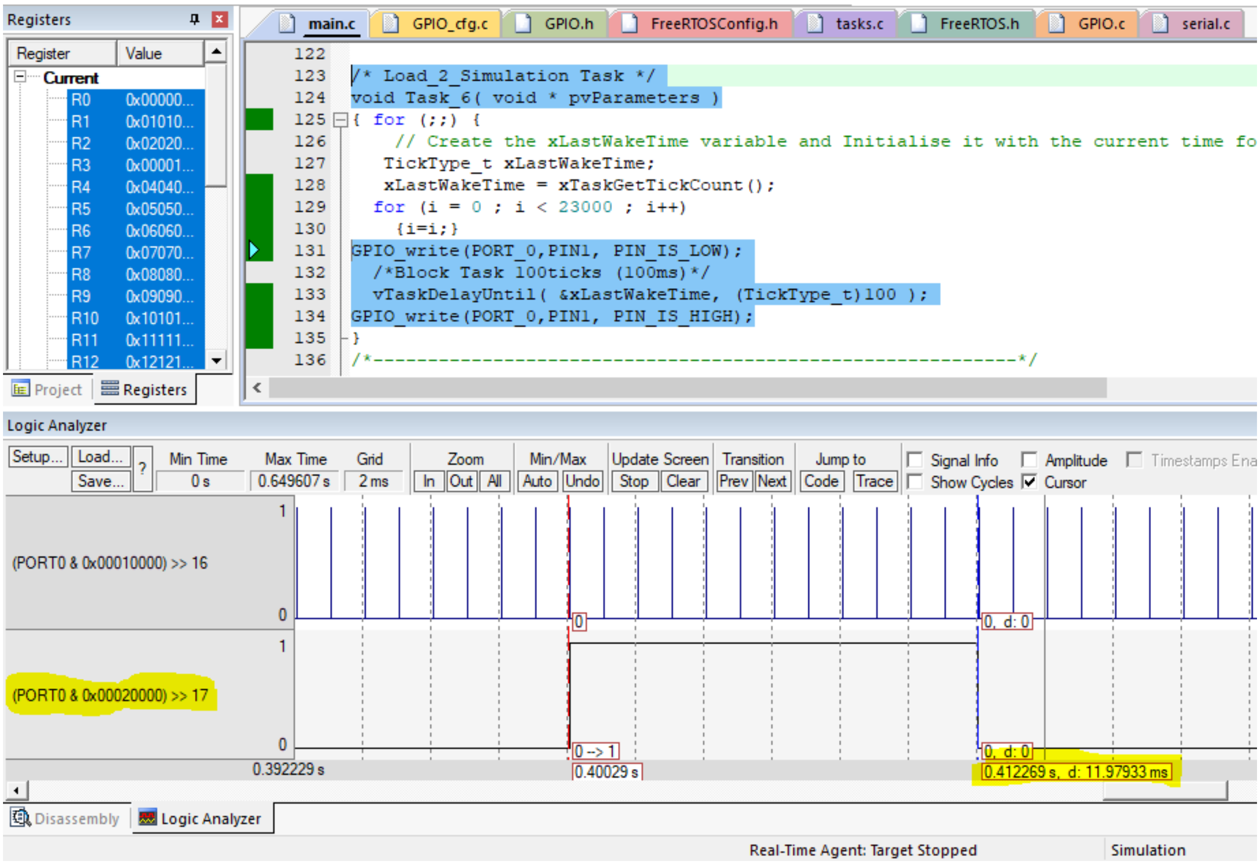
(PORT0 & 0x00020000) >> 17 — 0 --> 1 | 1, d: 1

(PORT0 & 0x00040000) >> 18 — 1 | 1, d: 0

84.86 ms | 0.100285 s | 0.20046 s, d: 0.100175 s

Open an existing document | Real-Time Agent: Target Stopped

→ Execution =~ 12ms



Register	Value
Current	
R0	0x00000...
R1	0x01010...
R2	0x02020...
R3	0x00001...
R4	0x04040...
R5	0x05050...
R6	0x06060...
R7	0x07070...
R8	0x08080...
R9	0x09090...
R10	0x10101...
R11	0x11111...
R12	0x12121...

```c
/* Load 2 Simulation Task */
void Task_6( void * pvParameters )
{ for (;;) {
   // Create the xLastWakeTime variable and Initialise it with the current time fo
   TickType_t xLastWakeTime;
   xLastWakeTime = xTaskGetTickCount();
  for (i = 0 ; i < 23000 ; i++)
    {i=i;}
GPIO_write(PORT_0,PIN1, PIN_IS_LOW);
  /*Block Task 100ticks (100ms)*/
  vTaskDelayUntil( &xLastWakeTime, (TickType_t)100 );
GPIO_write(PORT_0,PIN1, PIN_IS_HIGH);
}
/*-----------------------------------------------------------*/
```

Min Time 0 s | Max Time 0.649607 s | Grid 2 ms

(PORT0 & 0x00010000) >> 16 — 0 | 0, d: 0

(PORT0 & 0x00020000) >> 17 — 0 --> 1 | 0, d: 0

0.392229 s | 0.40029 s | 0.412269 s, d: 11.97933 ms

Real-Time Agent: Target Stopped | Simulation

CPU Load Calculation :

- CPU Load = total execution time of all tasks in one hyperperiod(cpu cusy tiime) / hyperperiod

= ((2*(13.9/1000)) + (2*(13.9/1000)) + (1* (7.5/1000)) + (5* (11.5/1000)) + (10 * (5)) + (1* (12))) / 100 = 0.621206
= 62.12 %