|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Group |  |  |  | ACC |
| Date of Lab Work |  |  |  |  |
| Name | 1 | Pratama Aji Nur Rahman | D400154003 |
|  | 2 | Amnaduny Akhara | D400154006 |
|  | 3 | Milzam Wafi Azhar | D400154007 | ACC Date : |
|  | 4 | Jeski Saputra | D400154009 | Revise Date : |

MODULE 1

MULTITASKING SYSTEM IN ARDUINO USING FREERTOS

1. PURPOSE
2. To know operating system in Arduino using FreeRTOS
3. To know the priority of multitasking system
4. TOOLS AND EQUIPMENT
5. Laptop
6. Arduino
7. BASIC THEORY

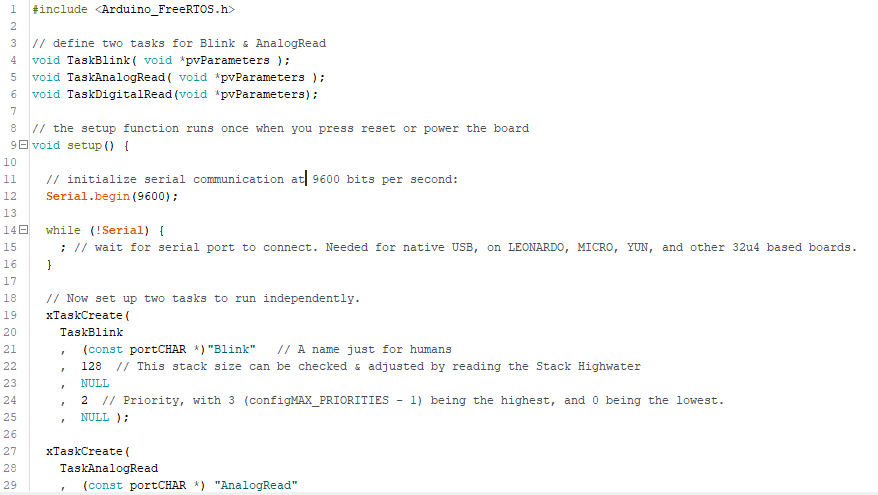
Most operating systems appear to allow multiple programs or threads to execute at the same time. This is called multi-tasking. In reality, each processor core can only be running a single program at any given point in time. A part of the operating system called the scheduler is responsible for deciding which program to run when, and provides the illusion of simultaneous execution by rapidly switching between each program.

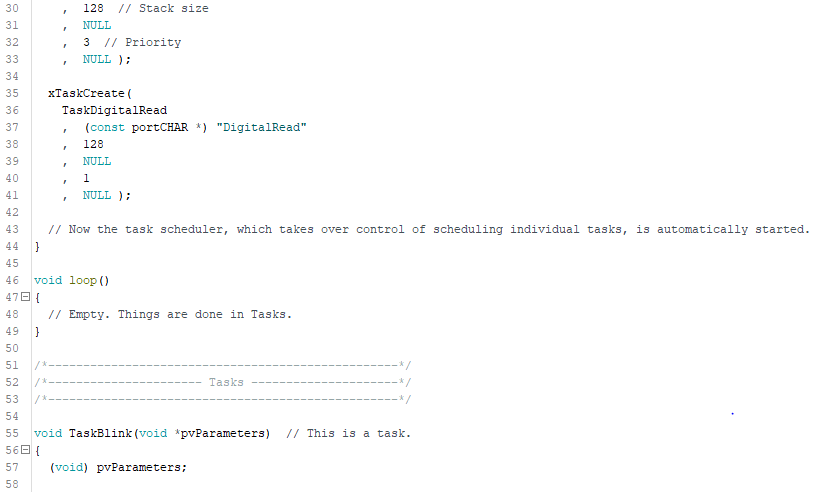
The scheduler in a Real Time Operating System (RTOS) is designed to provide a predictable (normally described as deterministic) execution pattern. This is particularly interesting for embedded systems, like the Arduino devices, as embedded systems often have real time requirements.

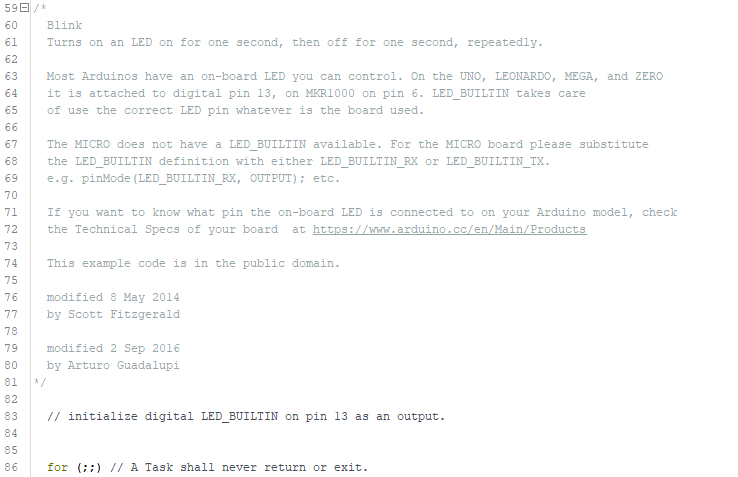
Traditional real time schedulers, such as the scheduler used in [FreeRTOS](http://www.freertos.org/RTOS.html), achieve determinism by allowing the user to assign a priority to each thread of execution. The scheduler then uses the priority to know which thread of execution to run next. In FreeRTOS, a thread of execution is called a Task.

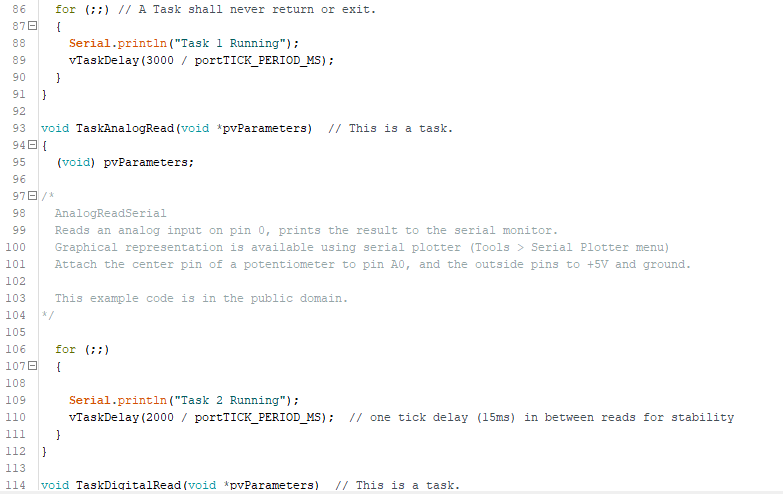
1. RESULT OF LAB WORK

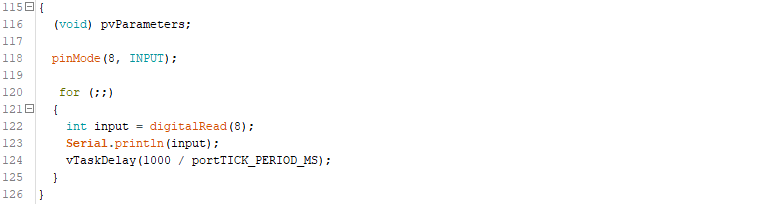
D.1 Script Program











C.4 Picture of Result

1. ANALYSIS

In this lab work using library of <Arduino\_FreeRTOS.h> to get started the FreeRTOS multitasking system. Line 3 to 6 is function which uses pointer. Insid void setup() is only for showing the serial communication in 9600 baudrate. xTaskCreate is to process the program and to choose the priority of the program. Higher value of priority makes higher priority. In FreeRTOS doesn’t need to fill program inside void loop(). Inside of xTaskCreate is filled the function of pointer in this case is Line 55 to 91, 93 to 111 and 114 to 125.

Void TaskBlink has program to show “Task 1 Running” with delay 3000 / portTICK\_PERIOD\_MS which mean (3000/15)\*15 = 3000ms. TICK inside library FreeRTOS is 15ms.

Void TaskAnalogRead has program to show “Task 2 Running” with delay (2000/15)\*15 = 2000ms.

Void TaskDigitalRead has program to read sensor in port digital 8 of Arduino with delay (1000/15)\*15 = 1000ms.

1. CONCLUSION
2. For the 1st priority is sensor LDR and 2nd priority is the remote IR because the main system is automatic system so the sensor is the 1st priority
3. Remote IR can’t interrupt LDR sensor because it will read and always read because it is in 1st priority then LDR sensor will active after 2nd priority works