

```

typedef enum {
    invalid,
    left,
    right,
    up,
    down,
} ornt_t;
QueueHandle_t ornt_queue;
ornt_t ornt;

void producer(void *p)
{
    while (1) {
        int xaxis = AS.getX();
        int yaxis = AS.getY();
        ornt = invalid;
        if((xaxis < -350) && (yaxis < 350) && (yaxis > -350)){
            ornt = right;
        }
        else if((xaxis > 350) && (yaxis < 350) && (yaxis > -350)){
            ornt = left;
        }
        if((yaxis < -350) && (xaxis < 350) && (xaxis > -350)){
            ornt = up;
        }
        else if((yaxis > 350) && (xaxis < 350) && (xaxis > -350)){
            ornt = down;
        }
        u0_dbg_printf("sending to queue\n");
        xQueueSend(ornt_queue, &ornt, portMAX_DELAY);
        u0_dbg_printf("sent to queue\n");
        vTaskDelay(1000);
    }
}

```

```

void consumer(void *p)
{
    while (1) {
        if(xQueueReceive(ornt_queue, &ornt, portMAX_DELAY)){
            switch(ornt){
                case 0:
                    u0_dbg_printf("invalid\n");
                    break;
                case 1:
                    u0_dbg_printf("left\n");
            }
        }
    }
}

```

```

        break;
    case 2:
        u0_dbg_printf("right\n");
        break;
    case 3:
        u0_dbg_printf("up\n");
        break;
    case 4:
        u0_dbg_printf("down\n");
        break;
    }
}
}
}

```

```

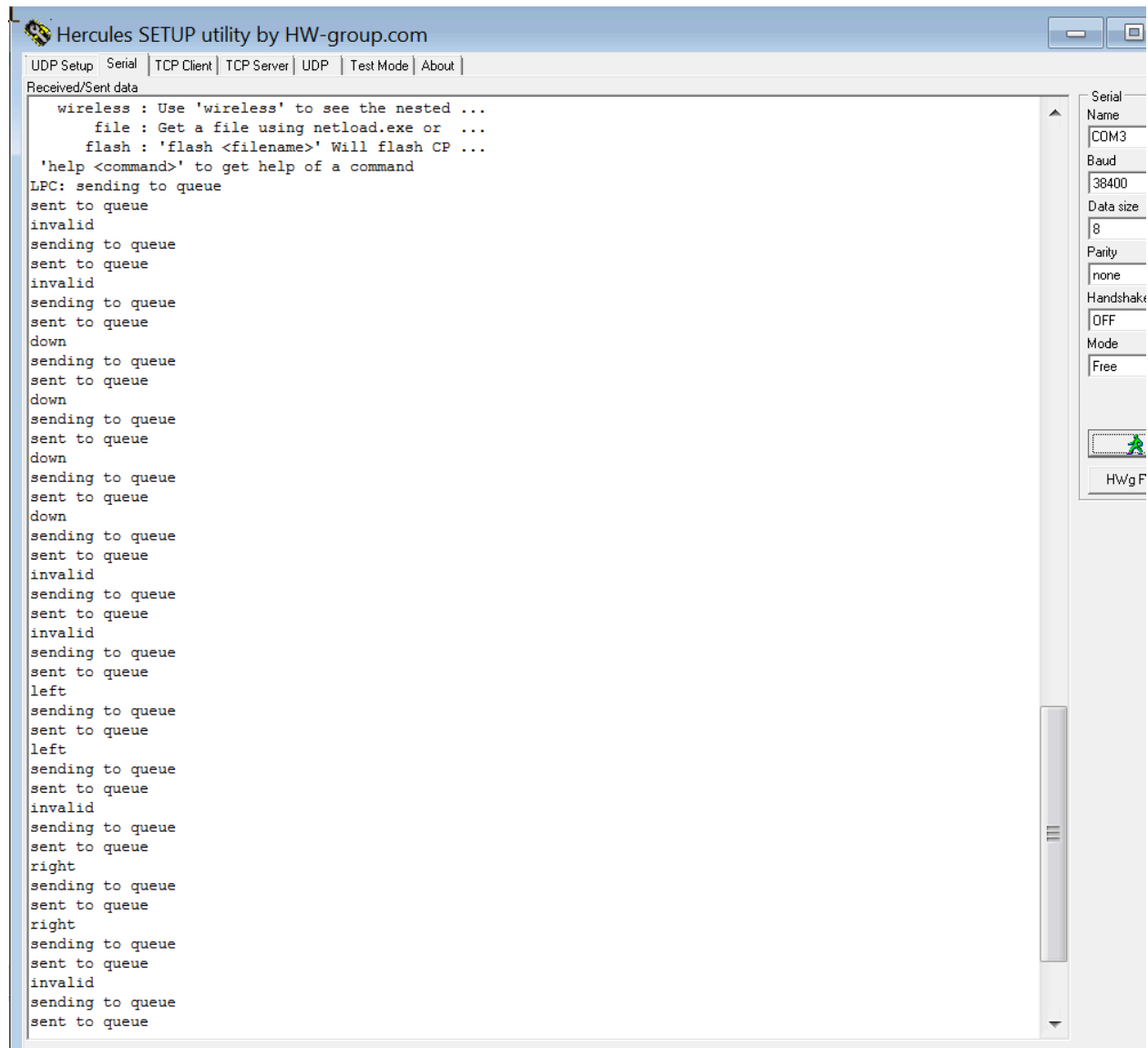
    ornt_queue = xQueueCreate(5, sizeof(int));
    xTaskCreate(producer, (const char*)"producer", STACK_BYTES(2048),
0, PRIORITY_HIGH, 0);
    xTaskCreate(consumer, (const char*)"consumer", STACK_BYTES(2048),
0, PRIORITY_HIGH, 0);

```

```

scheduler_add_task(new terminalTask(PRIORITY_HIGH));
scheduler_start();
vTaskStartScheduler();

```



With equal priority the orientation the task outputs orientation and then prints it has been sent. If the consumer task has a higher priority it waits for the queue for data then prints the message.

Block time being 0 would result incorrect correct as the queue needs time to relay data. The portmaxdelay allows proper timing for more accurate data. The block time on the xQueueRecieve allows the tasks to send data with a timing constraint, changing the timing could be helpful in debugging or speeding up the data transfer.