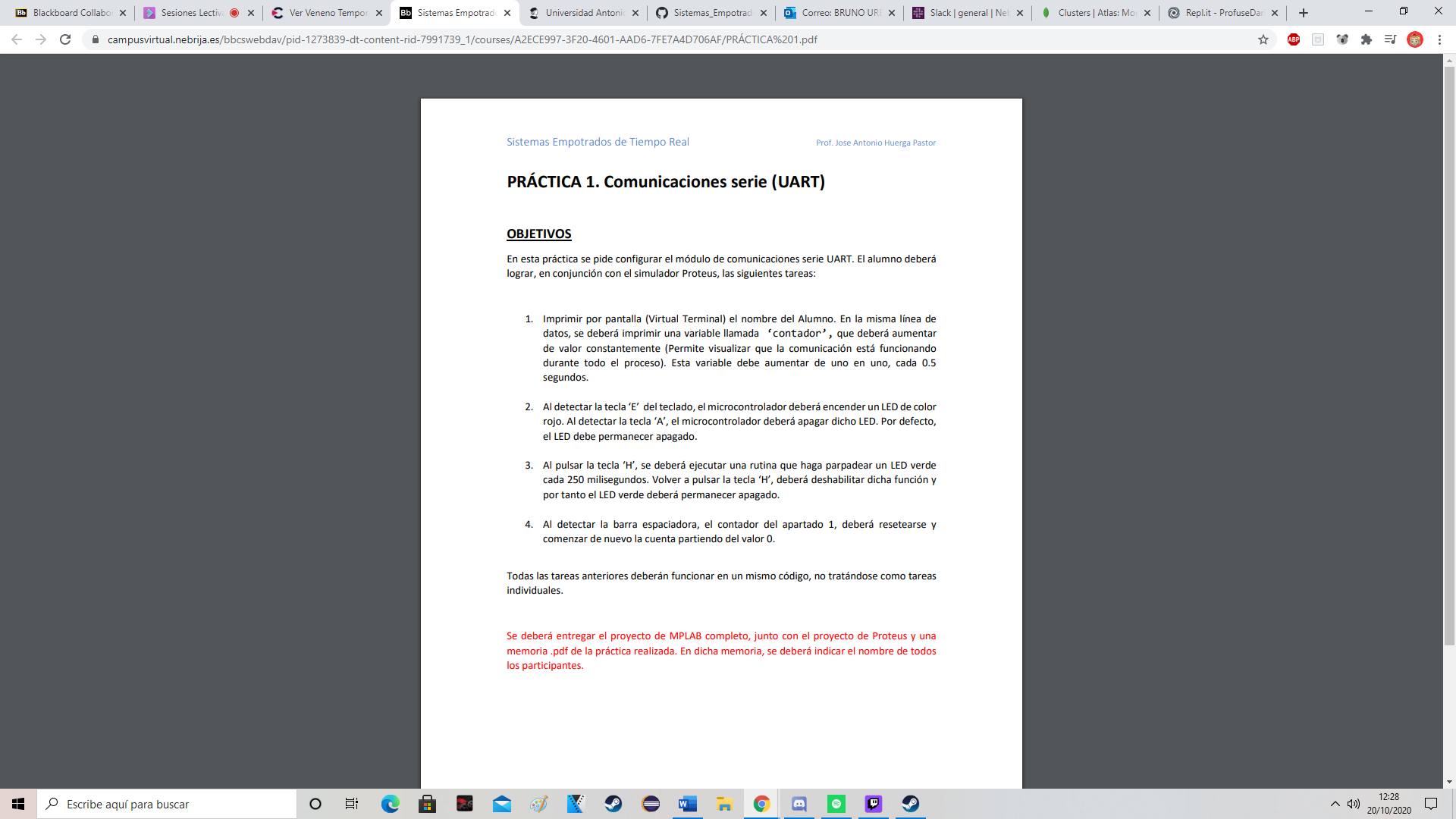
**Sistemas empotrados de tiempo real**



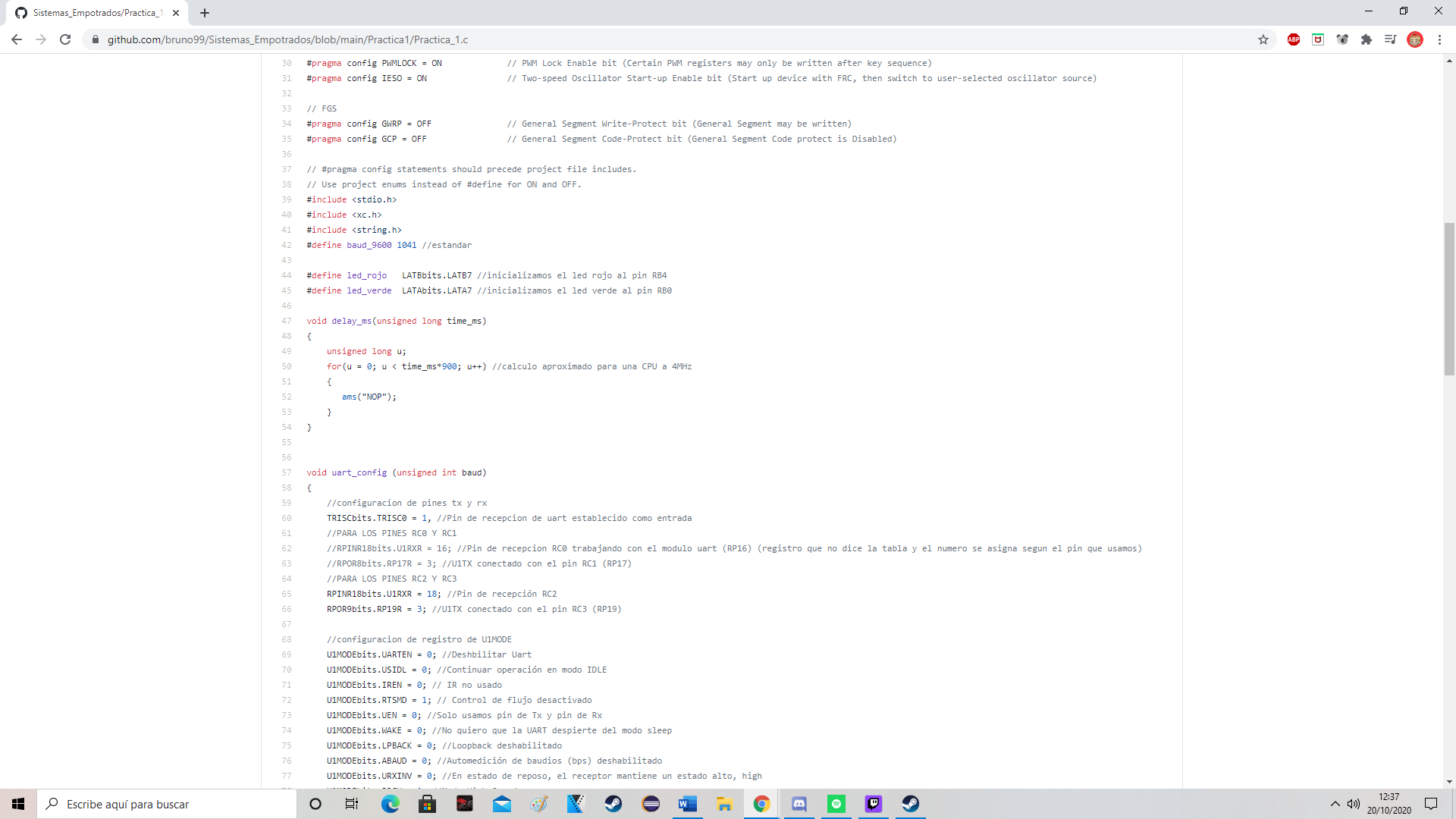
Bruno Urbán Alfaro

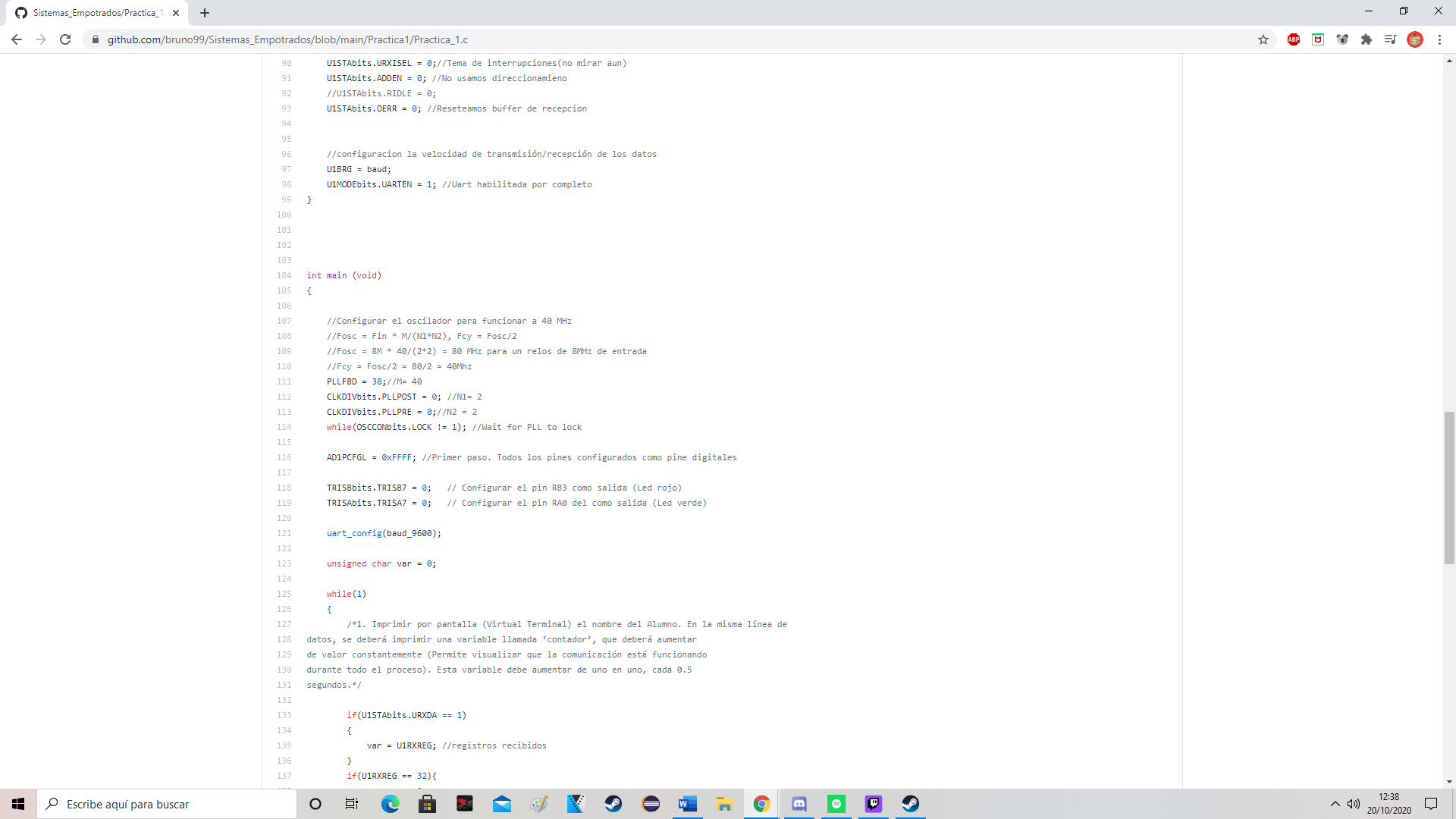
Esteban Vidiella

A partir de este circuito:

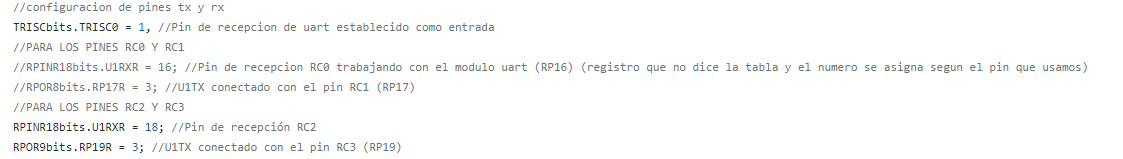
Diagrama, Esquemático

Descripción generada automáticamente

Definimos los nuevos pines para los leds



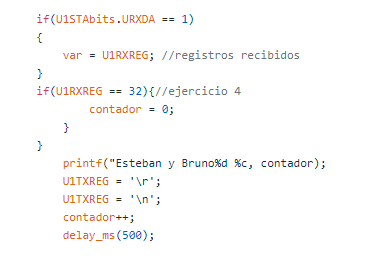
Configuramos los nuevos pines de uart (RC2 y RC3)



Hemos incluido: #include <stdbool.h> #include <stdio.h> #include <string.h>

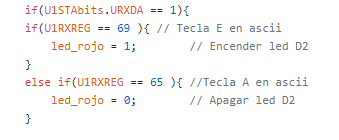
**1. Imprimir por pantalla (Virtual Terminal) el nombre del Alumno. En la misma línea de datos, se deberá imprimir una variable llamada ‘contador’, que deberá aumentar de valor constantemente (Permite visualizar que la comunicación está funcionando durante todo el proceso). Esta variable debe aumentar de uno en uno, cada 0.5 segundos.**

URXDA a 1 porque el buffer tiene recepción de datos.

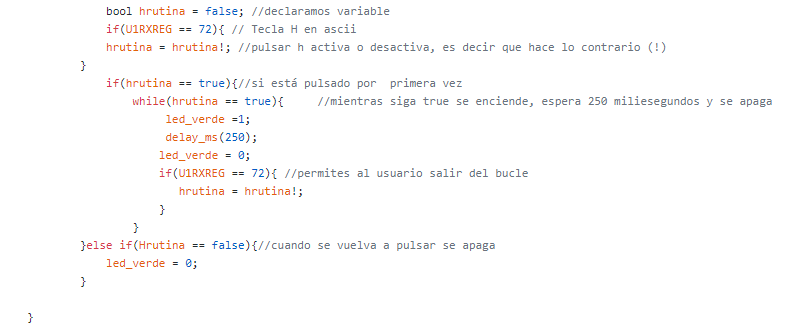


Inicializamos una variable contador (var) a 0 . Se imprimen nuestros nombres usando printf (usamos esta función porque sprtint nos daba errores con el buffer) con el contador aumentando cada 500ms

**2. Al detectar la tecla ‘E’ del teclado, el microcontrolador deberá encender un LED de color rojo. Al detectar la tecla ‘A’, el microcontrolador deberá apagar dicho LED. Por defecto, el LED debe permanecer apagado.**

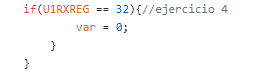


Como el 69 en ascii es ‘E’ y el 65 es ‘A’, nuestro programa encenderá el led rojo si detecta esta primera y lo apagará si detecta la segunda.

**3. Al pulsar la tecla ‘H’, se deberá ejecutar una rutina que haga parpadear un LED verde cada 250 milisegundos. Volver a pulsar la tecla ‘H’, deberá deshabilitar dicha función y por tanto el LED verde deberá permanecer apagado.**

Inicializamos un booleano (hrutina) que nos dice si H está activado o no. Si se pulsa H, el bool cambia de estado. Si H es true entramos en un bucle que enciende y apaga el led cada 250ms para que el led parpadee. Hemos metido de nuevo la condición de la tecla H porque si no sería imposible que el usuario saliera del bucle. Una vez el usuario pulsa H y el bool es false, el led se vuelve a apagar permanentemente.

**4. Al detectar la barra espaciadora, el contador del apartado 1, deberá resetearse y comenzar de nuevo la cuenta partiendo del valor 0.**



En el ejercicio 1 se ve que hemos puesto la condición en la barra espaciadora (32 en ascii) para que el contador vuelva a 0.

**\*La línea 117 da error pese a estar escrita igual que en clase\* -🡪** AD1PCFGL = 0xFFFF;