

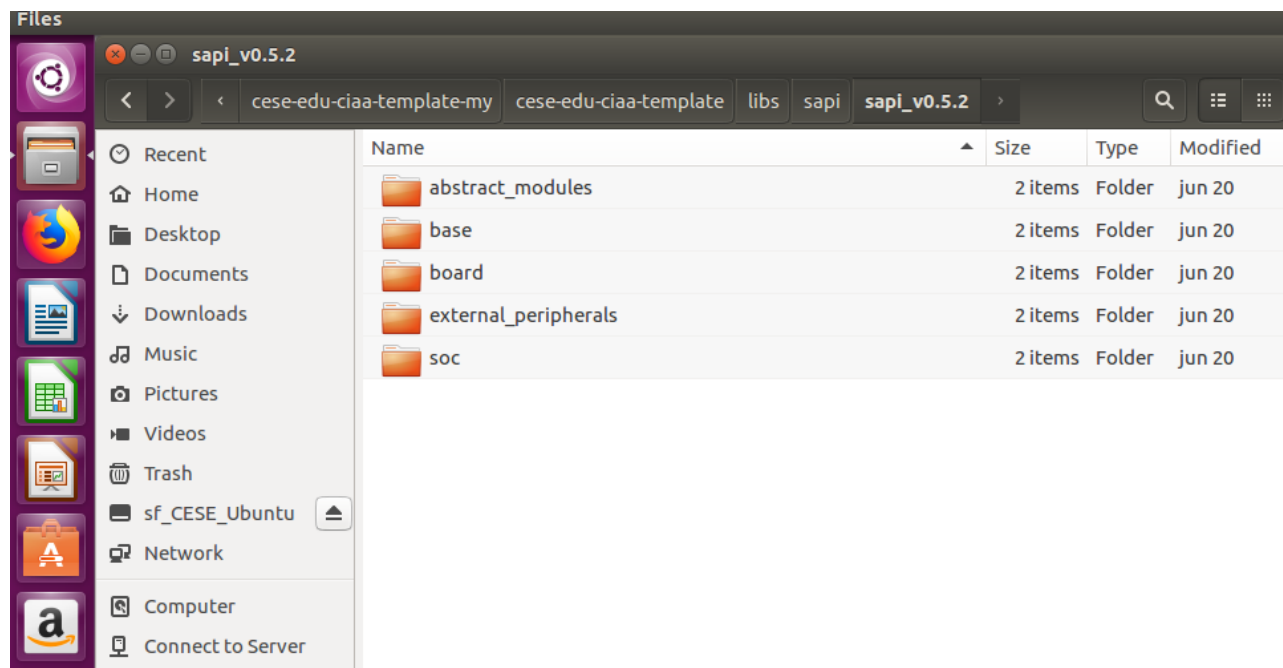
5.08.2019 CESE_2019_PROY-FINAL

R. Oliva

Instructivo para uso de Ejemplos sapi3c basados en sAPI (de Mg. Ing. Eric Pernia) para CIAA

A) Se toma como Referencia la Estructura de la sAPI 0.5.2 (04.2019) para la CIAA / EDU-CIAA

V0.5.2 – dividida como sigue (/sapi está dentro de /libs):



B) Se adapta esta estructura para su utilización en la Placa Prototipo CL3, basada en un STM32F411. Si bien se utiliza gcc, y un editor/IDE AC6 abierto basado en Eclipse, la adaptación de las bibliotecas y la compilación de proyectos requiere significativos cambios, que se describen en la Memoria del Trabajo Final. El desarrollo se realizó partiendo del hardware prototipo a través del software CubeMX de ST, y la utilización mixta de bibliotecas LL y HAL de ST para la línea STM32F4 (Cortex M4F)

C) El presente instructivo sólo pretende mostrar la estructura de los ejemplos y la biblioteca preliminar sapi3c, cuyo objetivo es tener la facilidad de programación que ofrece la sAPI original para la CIAA.

D) En la presente versión del documento se describen dos de los ejemplos muy elementales, pero que ilustran la estructura de la biblioteca de programación:

- Ej 01 / Simple blinky

- Ej 02 / Lectura de Teclado y mensaje por UART Terminal.

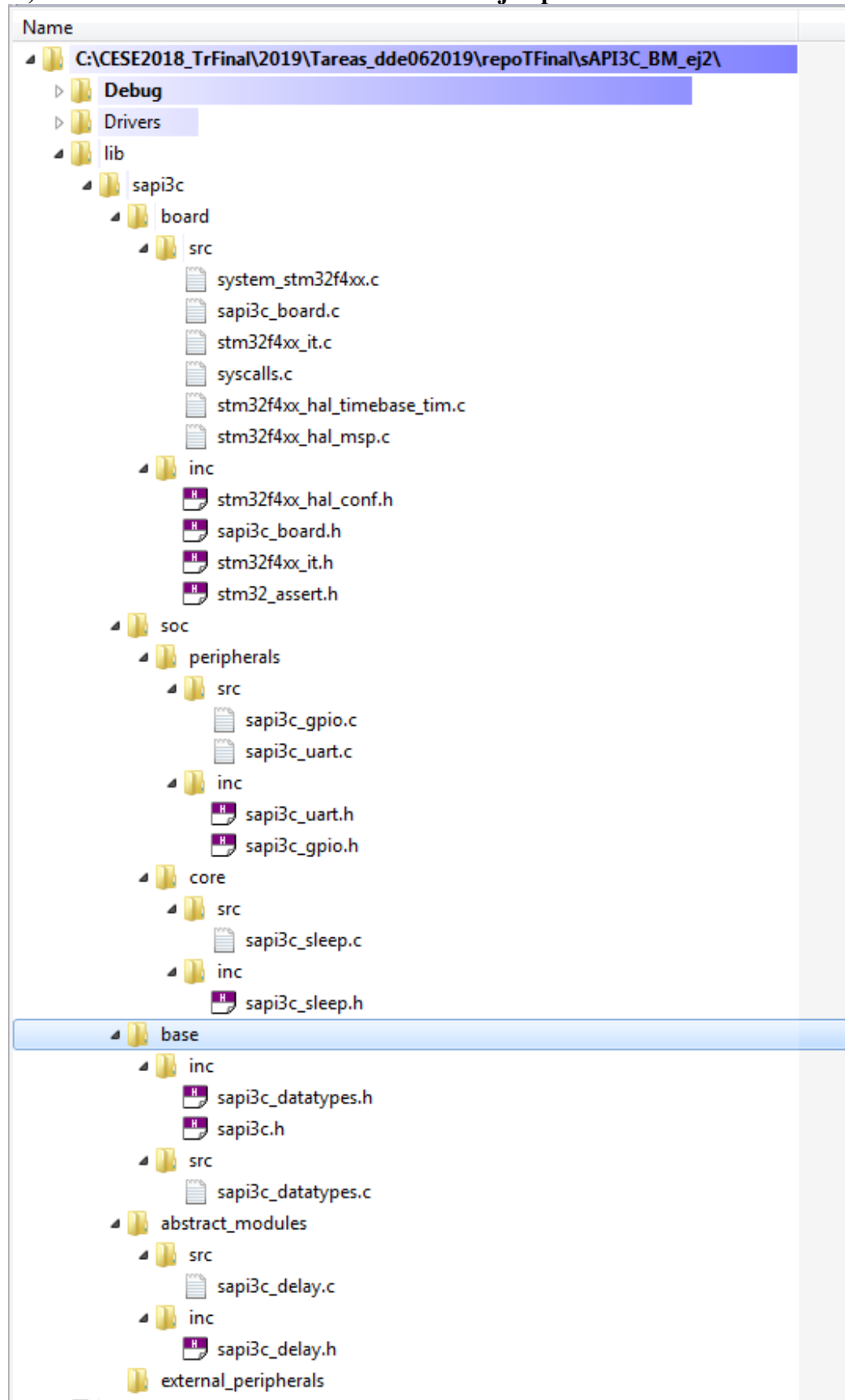
https://github.com/rafaeloliva/CESE_TrabFinal2019/tree/master

E) Para compilar y modificar los ejemplos se requiere instalar la herramienta libre AC6/System Workbench desde ST.com ó :

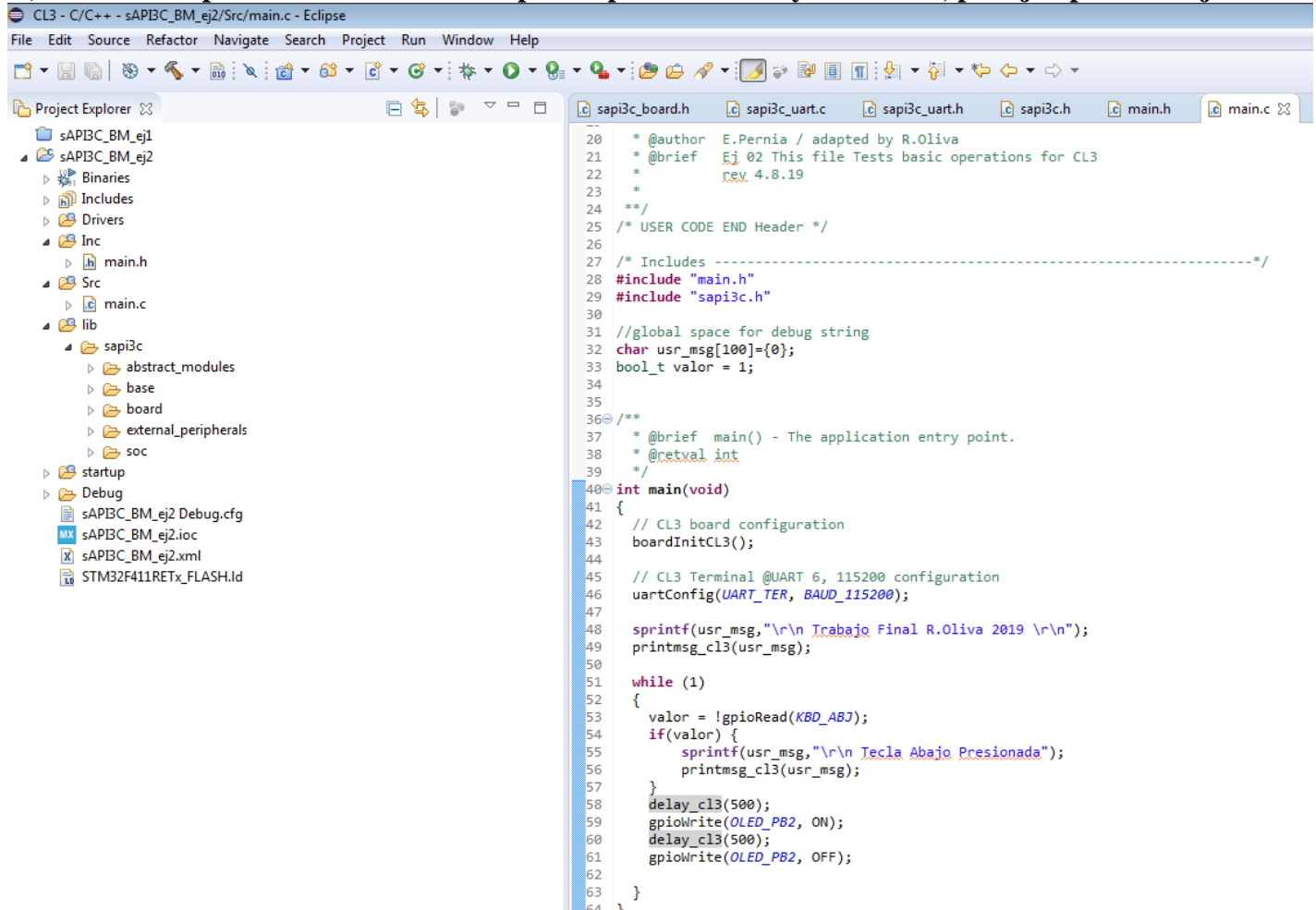
<http://www.openstm32.org/System%2BWorkbench%2Bfor%2BSTM32>

F) Al descargar del presente repositorio, en los ejemplos aparecen dos archivos .project y .cproject. Se debe asociar dichos archivos al AC6/SystemWorkbench previamente instalado. Luego dar doble click en cualquiera de ambos .project (del ej01 o del ej02) y elegir un directorio de trabajo (Workspace) de AC6 para instalarlos.

G) Estructura de la Biblioteca usada en los ejemplos:



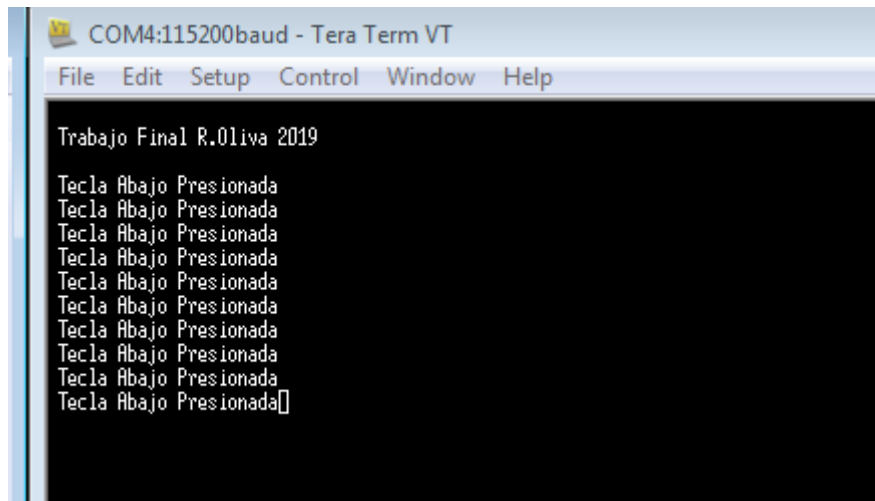
H) Al usuario aparece una interfase compuesta por un main.c y un main.h, por ejemplo en el Ej02:



The screenshot shows the Eclipse IDE with the project 'sAPBC_BM_ej2' open. The Project Explorer on the left shows the file structure, including 'main.h' and 'main.c' in the 'Src' folder. The main.c file is open in the editor, showing the following code:

```
20  * @author E.Pernia / adapted by R.Oliva
21  * @brief Ej 02 This file Tests basic operations for CL3
22  * rev 4.8.19
23  *
24  **/
25  /* USER CODE END Header */
26
27  /* Includes -----*/
28  #include "main.h"
29  #include "sapi3c.h"
30
31  //global space for debug string
32  char usr_msg[100]={0};
33  bool_t valor = 1;
34
35
36 /**
37  * @brief main() - The application entry point.
38  * @retval int
39  */
40 int main(void)
41 {
42     // CL3 board configuration
43     boardInitCL3();
44
45     // CL3 Terminal @UART 6, 115200 configuration
46     uartConfig(UART_TER, BAUD_115200);
47
48     sprintf(usr_msg, "\r\n Trabajo Final R.Oliva 2019 \r\n");
49     printmsg_cl3(usr_msg);
50
51     while (1)
52     {
53         valor = !gpioRead(KBD_ABJ);
54         if(valor) {
55             sprintf(usr_msg, "\r\n Tecla Abajo Presionada");
56             printmsg_cl3(usr_msg);
57         }
58         delay_cl3(500);
59         gpioWrite(OLED_PB2, ON);
60         delay_cl3(500);
61         gpioWrite(OLED_PB2, OFF);
62     }
63
64 }
```

Este Ej02 imprime un mensaje inicial, y luego en el lazo del LED otro si detecta presionada la tecla KBD_ABJ = F3



I) Foto de los ensayos de la Placa

