UNIVERSIDAD
COMPLUTENSE
MADRID

Proyecto de Innovación

Convocatoria 2021/2022

Nº de proyecto: 386

Entrenador RISC-V (RISC-V Trainer)

Responsable del proyecto:
Juan Carlos Fabero Jiménez

Facultad de Informática
Departamento:
Arquitectura de Computadores y Automática

Índice

1. Objetivos propuestos en la presentación del proyecto

RISC-V es una ISA (*Instruction Set Architecture*) abierta que permitirá el futuro desarrollo de microprocesadores libres y abaratar los costes de diseño y producción. Desde su propuesta, en 2010 por parte de la Universidad de Berkeley, numerosas empresas como Western Digital, Espressif, ZTE, y SiFive, entre otras, han desarrollado y sacado al mercado chips basados en RISC-V.

En la Facultad de Informática de la UCM se imparten varias asignaturas donde se introducen los conceptos de las microarquitecturas de los procesadores actuales, como son Fundamentos de Computadores II (de primer curso) y Estructura de Computadores (segundo curso). En estas asignaturas se describe la arquitectura RISC-V, por lo que puede resultar conveniente el uso de placas basadas en esta arquitectura para las prácticas. En otras asignaturas, como Programación de Sistemas y Dispositivos, se emplean placas basadas en microprocesador para el control de diversos sensores y actuadores.

Durante el presente curso académico se han realizado algunas experiencias con placas basadas en RISC-V. Aunque ya existen en el mercado algunas de estas placas, se echa en falta la periferia adecuada para la realización de las prácticas de entrada/salida que permitan a los estudiantes afianzar los conocimientos impartidos en la teoría.

Otro problema que presentan las placas actuales es la falta de un sistema de depuración. Nos propusimos incorporar dicha facilidad al diseño mediante un conversor JTAG integrado.

Al emplear microprocesadores RISC-V, un mercado actualmente en expansión, donde ya se ha mencionado que empieza a haber disponibilidad de placas de bajo coste, podemos conseguir claramente aumentar el interés de los alumnos por la asignatura, facilitar el aprendizaje autónomo de la misma e incentivar la creatividad, al permitir aplicar los conocimientos adquiridos a un enorme conjunto de posibilidades.

En los laboratorios de la Facultad de Informática se dispone de medios apropiados para permitir el desarrollo de un entorno donde programar y utilizar múltiples dispositivos de entrada/salida. Esto sin embargo no es nada trivial y requiere el esfuerzo de un grupo de profesores con amplios conocimientos sobre diseño de sistemas para la puesta a punto del mismo.

Por una parte, disponemos de placas de prototipado que se pueden utilizar para realizar un sistema empotrado básico, compuesto por un procesador y los diferentes dispositivos de entrada/salida ya mencionados. Además, se dejarán conectores compatibles con Arduino. De esta forma será posible programar y utilizar múltiples dispositivos de entrada/salida y que los alumnos, usando la documentación que les proporcionamos, puedan diseñar sistemas complejos, utilizar la amplia oferta de “escudos” (*shields*) de ampliación disponibles en el mercado y ejercitar su imaginación y creatividad para aprender de forma fácil y atractiva cómo funcionan.

En particular, los objetivos concretos de este proyecto son los siguientes:

- Desarrollo de un entorno que permita programar y probar distintos tipos de periféricos para favorecer el aprendizaje autónomo de los alumnos. Este entorno se desarrollará en una placa basada en un microprocesador RISC-V, y estará formado por el procesador, buses, una conexión serie para utilizar el terminal y el entorno de desarrollo y depuración de las aplicaciones.
- Desarrollo de conectores “Plug & Play” para la conexión al entorno de los distintos dispositivos.
- Ampliación de la placa para incorporar los sensores analógicos, los dispositivos controlados por el bus I²C, visualizadores LED de 7 segmentos, pulsadores, etc.
- Desarrollo de manuales que faciliten al alumno la comprensión del modelo de programación de cada uno de los periféricos mencionados, y otros tales como sensores de luz, temperatura, proximidad y contacto, altavoces, conversores analógico/digital y digital/analógico, moduladores de pulsos, expansores de entrada/salida, bus SPI, etc.
- Integración con el entorno de programación y depuración (PlatformIO, CodeStudio...) para facilitar el desarrollo de las prácticas.
- Desarrollo de prácticas básicas que permitan al alumno asimilar el funcionamiento de los periféricos anteriormente descritos.
- Conseguir que el alumno sea capaz de desarrollar un sistema empotrado completo a partir de los conocimientos adquiridos durante las prácticas.
- Favorecer la iniciativa del alumno para que éste aporte su creatividad.

2. Objetivos alcanzados

Podemos decir que se han cumplido todos los objetivos propuestos en el proyecto. A continuación pasamos a detallar los hitos conseguidos:

- Se ha integrado un entorno de programación y depuración que permite el desarrollo de aplicaciones para la arquitectura RISC-V.
- Se ha seleccionado un SoC (*System on Chip*) y se ha diseñado una placa de circuito impreso que permite, por un lado, el uso de los sensores y actuadores incorporados

y, además, al estar dotada de conectores de expansión, incorporar de manera sencilla nuevos elementos *hardware* que permiten a los estudiantes dar rienda suelta a su creatividad.

- Se han escrito las librerías necesarias para la utilización de los periféricos incluidos en la placa, así como ejemplos sencillos que demuestran su uso.

3. Metodología empleada en el proyecto

La metodología que se ha seguido durante el desarrollo de este Proyecto de Innovación ha constado de los siguientes pasos:

- Estudio de las diferentes placas y alternativas de SoC con arquitectura RISC-V existentes en el mercado.

  Finalmente se optó por el SoC ESP32-C3, de la empresa Espressif, por varios motivos:

  - Buena documentación de la arquitectura interna. Esto es fundamental para poder realizar prácticas adecuadas a las asignaturas de Estructura de Computadores y Programación de Sistemas y Dispositivos.
  - Incorpora un interfaz USB-JTAG, lo que simplifica el diseño de la placa de circuito impreso.
  - Bajo coste (por debajo de 5 euros cada unidad).
  - Integración con el entorno de programación (IDE) de Arduino y posibilidad de depuración mediante GDB (*GNU Debugger*).

- Integración de la parte *software*. Esta parte incluyó la adecuación del IDE de Arduino al ESP32-C3 y la configuración y manejo del depurador GDB.

- A partir de una placa ya construida, en concreto la Waveshare ESP-C3-32S-Kit, se desarrollaron las librerías básicas para el manejador de periféricos.

- Puesta en funcionamiento del controlador de periféricos I²C.

- Elaboración de los manuales de uso de los dispositivos y del entorno de desarrollo.

- Preparación de las prácticas que deben realizar los estudiantes.

- Propuesta de proyectos básicos que abran las puertas a la creatividad de los estudiantes.

4. Recursos humanos

Todos los miembros son expertos diseñadores de sistemas hardware.

- Juan Antonio Clemente ha impartido las asignaturas relacionadas: Laboratorio de Tecnología de Computadores (LTC), Laboratorio de Estructura de Computadores (LEC) y Ampliación de Estructura de Computadores (AEC), de la titulación Ïngeniero en Informática", así como las asignaturas: Tecnología y Organización de Computadores (TOC), Fundamentos de Computadores (FC) y Diseño de Sistemas Empotrados (DSE) del nuevo Grado en Ingeniería de Computadores. En el proyecto ha colaborado en el diseño de la placa.

- Juan Carlos Fabero tiene una amplia experiencia en la asignatura de Sistemas Operativos (SO) y Estructura de Computadores (EC), ambas fundamentales para el desarrollo de los objetivos de este proyecto. Fue profesor durante 8 años de la asignatura Laboratorio de Estructura de Computadores (LEC), impartida en la antigua titulación de Ingeniería Superior en Informática, donde se utilizaba un microprocesador Motorola 68K para el control de diversos dispositivos de entrada/salida. Se ha encargado de elegir el SoC e integrar los entornos de desarrollo y depuración.

- Daniel León es estudiante de doctorado en la Facultad de Informática de la UCM. Actualmente desarrolla su tesis doctoral sobre impacto de la radiación en dispositivos RISC-V comerciales para misiones espaciales, dentro del grupo GHADIR de DACYA/UCM. Ha participado en el proyecto RVFpga, (Art.83 - ArTeCS/UCM) para la creación de materiales docentes sobre RISC-V en FPGA y forma parte del grupo de ChipsAlliance para BlockChain con RISC-V. Además, es el profesor coordinador de las asignaturas de Electrónica/Tecnología de Computadores (ETC) y de Arquitectura y Organización de Computadores (AOC) en el grado de Ingeniería Informática de la Universidad Francisco de Vitoria. Previamente a su trayectoria docente, ha diseñado sistemas embebidos, principalmente para instrumentos musicales profesionales. Se ha encargado del diseño de la placa.

- Hortensia Mecha ha sido la profesora coordinadora de la asignatura Laboratorio de Tecnología de Computadores (LTC), donde se utilizaba como base para las prácticas el *hardware* reconfigurable. En esta asignatura se desarrollaron un conjunto de prácticas de diseño sobre FPGA. También fue la profesora de las asignaturas Diseño de Circuitos Integrados I y Diseño de Circuitos Integrados II de la Ingeniería en Informática. En la actualidad es la profesora de la asignatura de Sistemas Empotrados (SE) y Tecnología y Organización de Computadores (TOC), donde también se utiliza el *hardware* reconfi-

gurable para la realización de las prácticas. Además, en SE se utiliza una placa de expansión, desarrollada por varios de los profesores de este proyecto, en el marco de otros dos proyectos de innovación educativa. Esta placa de expansión proporciona a la FPGA distintos dispositivos de E/S, como LED de colores, conversores analógicos digitales, un motor paso a paso, una matriz de puntos, un LCD, un emisor/receptor de infrarrojos y un controlador de bus I²C. Ha colaborado en la realización de las prácticas.

- José Manuel Mendías Cuadros ha sido profesor de las asignaturas Programación de Sistemas y Dispositivos (PSyD), y Diseño Automático de Sistemas (DAS), ambas impartidas en el Grado en Ingeniería Informática. Tiene amplia experiencia en la programación de procesadores ARM. Ha sido el responsable de la realización de las prácticas para los estudiantes.

- Daniel Mozos ha sido profesor de un buen número de asignaturas relacionadas con el diseño de sistemas: Estructura de Computadores (EC), Tecnología de Computadores (LTC), Electrónica (E), y cursos de máster en Diseño de Sistemas Empotrados y Hardware Dinámicamente Reconfigurable dentro del Máster en Investigación en Informática. Ha colaborado en la integración de los entornos de desarrollo y depuración.

5. Desarrollo de las actividades

La mayor parte de las actividades se han desarrollado según el cronograma previsto, excepto las referentes al diseño de la placa de circuito impreso. Esto último se ha debido a la actual escasez de componentes electrónicos, lo que retrasó el la adquisición de los SoC ESP32-C3. Sin este componente no era posible disponer de su “huella” real, lo que paralizó el diseño final de la placa y, por consiguiente, su fabricación. Por ello, debimos solicitar la prórroga en la entrega de la memoria final del proyecto.

Sin embargo, dado que disponíamos de una placa de desarrollo Waveshare ESP-C3-32S-Kit (ver la Figura 1), pudimos desarrollar las actividades relacionadas con los entornos de programación y depuración, así como la preparación de las prácticas para los estudiantes.

Una vez que llegaron los SoC ESP32-C3 (Figura 2) se pudo completar el proyecto con el diseño y fabricación de la placa de entrenamiento.

5.1. Resultados y productos

Los resultados de este proyecto han sido los siguientes:

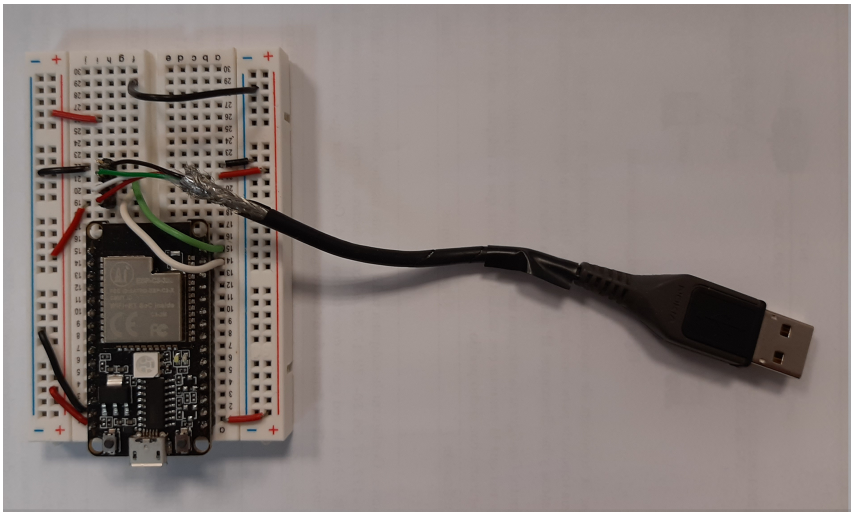

FIGURA 1: Placa de desarrollo Waveshare ESP-C3-32S-Kit basada en Espressif ESP32-C3 con interfaz USB para programación y depuración.

FIGURA 2: *System on Chip* (SoC) ESP32-C3 de Espressif.

- 26 placas con el microcontrolador ESP32-C3-WROOM-02 y periféricos asociados: LED, emisor/receptor de IR, sensor de luminosidad analógico mediante LDR, pantalla de visualización OLED, sensor de temperatura, presión y humedad BME280, y buses de expansión I²C y SPI.
- Librerías básicas en C para el uso de los periféricos.
- Entorno de desarrollo para compilación, carga, ejecución y depuración de los programas.
- Ejemplos de prácticas que hacen uso de los periféricos.

En la Figura 3 puede verse el esquemático del diseño *hardware* con los periféricos incorporados. En la Figura 4 mostramos el diseño final de la placa, ya lista para montar los componentes y, por último, en la Figura 5, el resultado final.

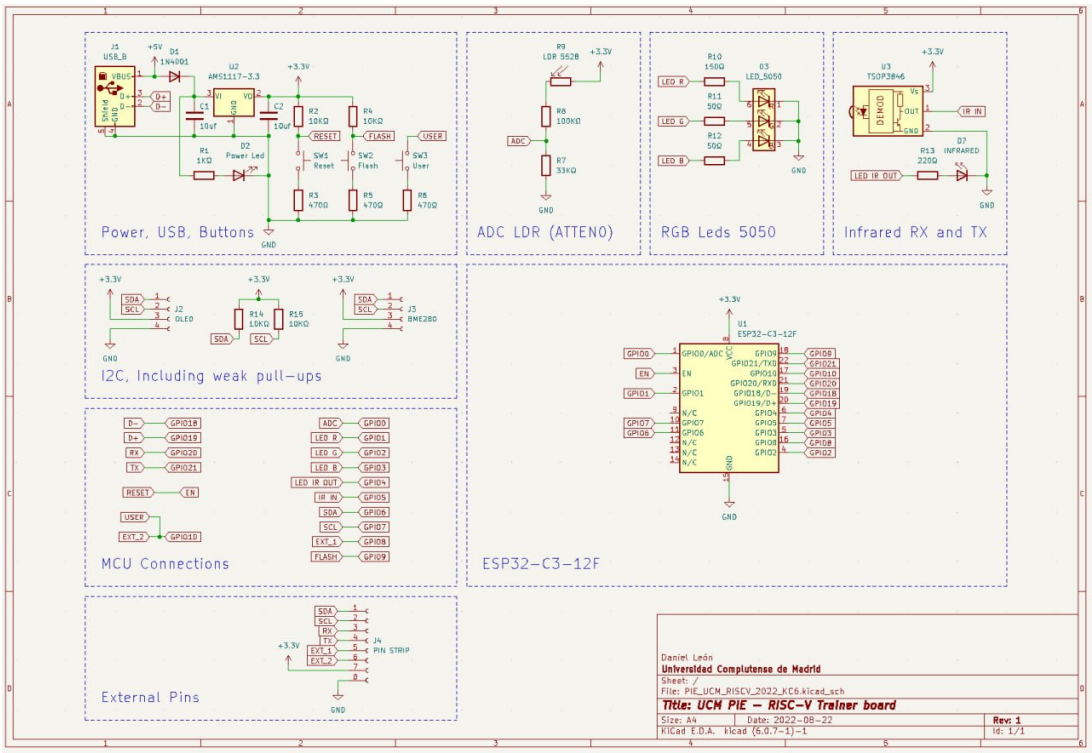

FIGURA 3: Esquemático de la placa de entrenamiento.

6. Anexos

Para finalizar, se incluyen ejemplos de una sesión de programación y depuración de una práctica sencilla.

En la Figura 6 se puede ver cómo se emplea el entorno de programación de Arduino para desarrollar el programa que se subirá a la placa. El mismo entorno se utiliza para subir el ejecutable al microcontrolador.

En la Figura 7 se muestra una sesión de depuración mediante GDB (GNU Debugger). Se permite la ejecución paso a paso, la definición de puntos de ruptura y todos los mecanismos adecuados para una correcta depuración.

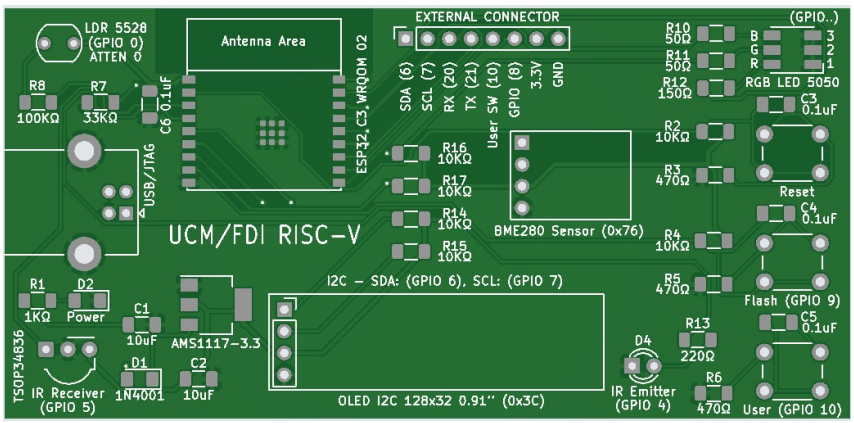

FIGURA 4: Diseño de la placa de circuito impreso.

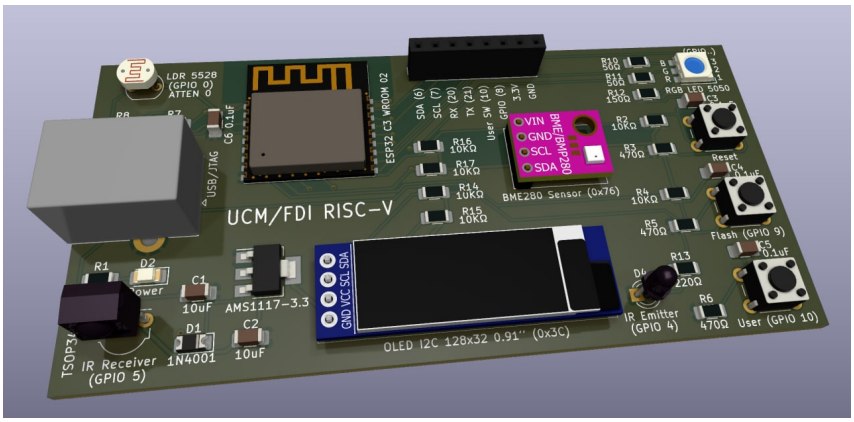

FIGURA 5: Aspecto final de la placa de entrenamiento.

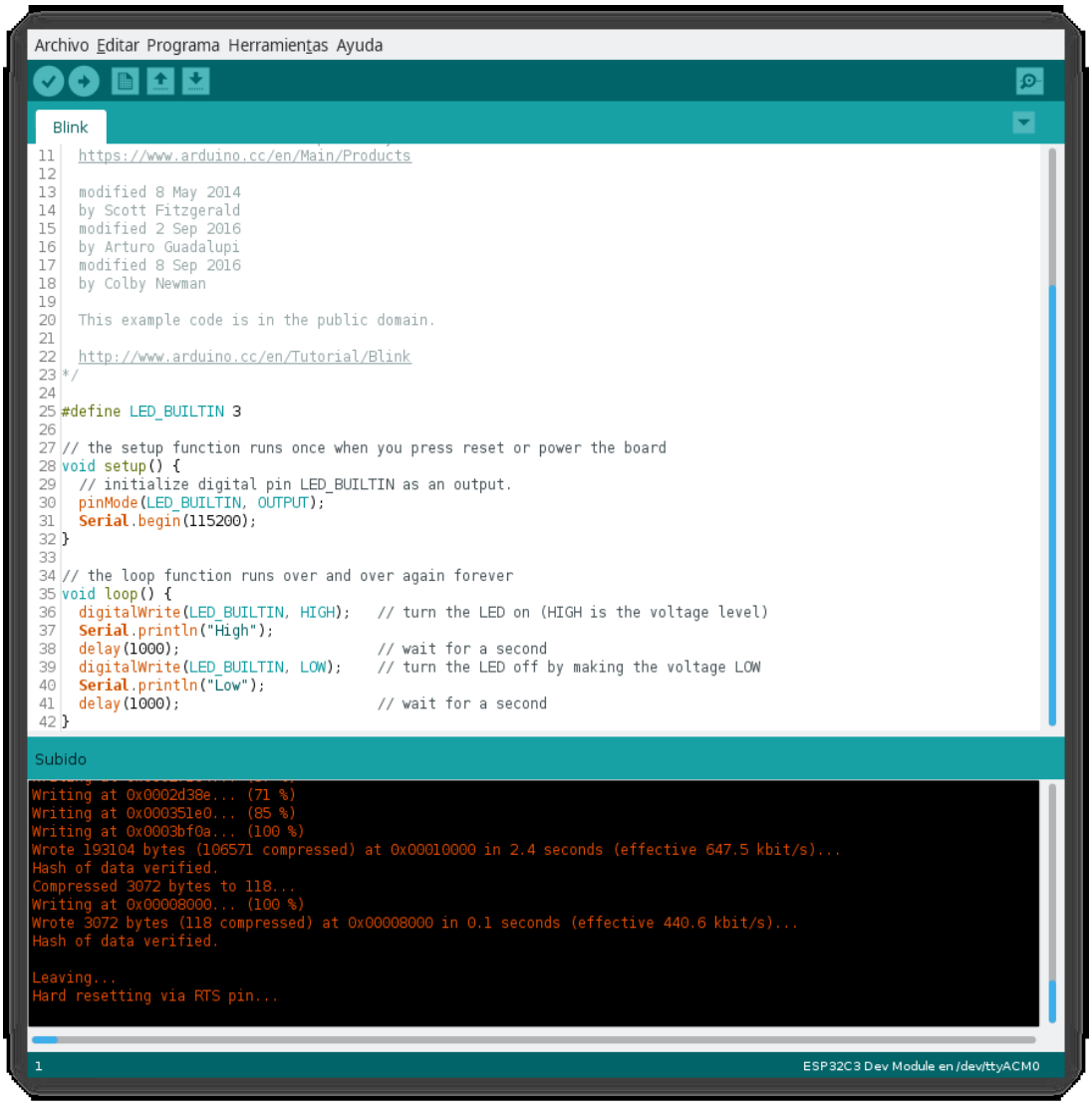

FIGURA 6: Desarrollo del *software* en el entorno de Arduino.

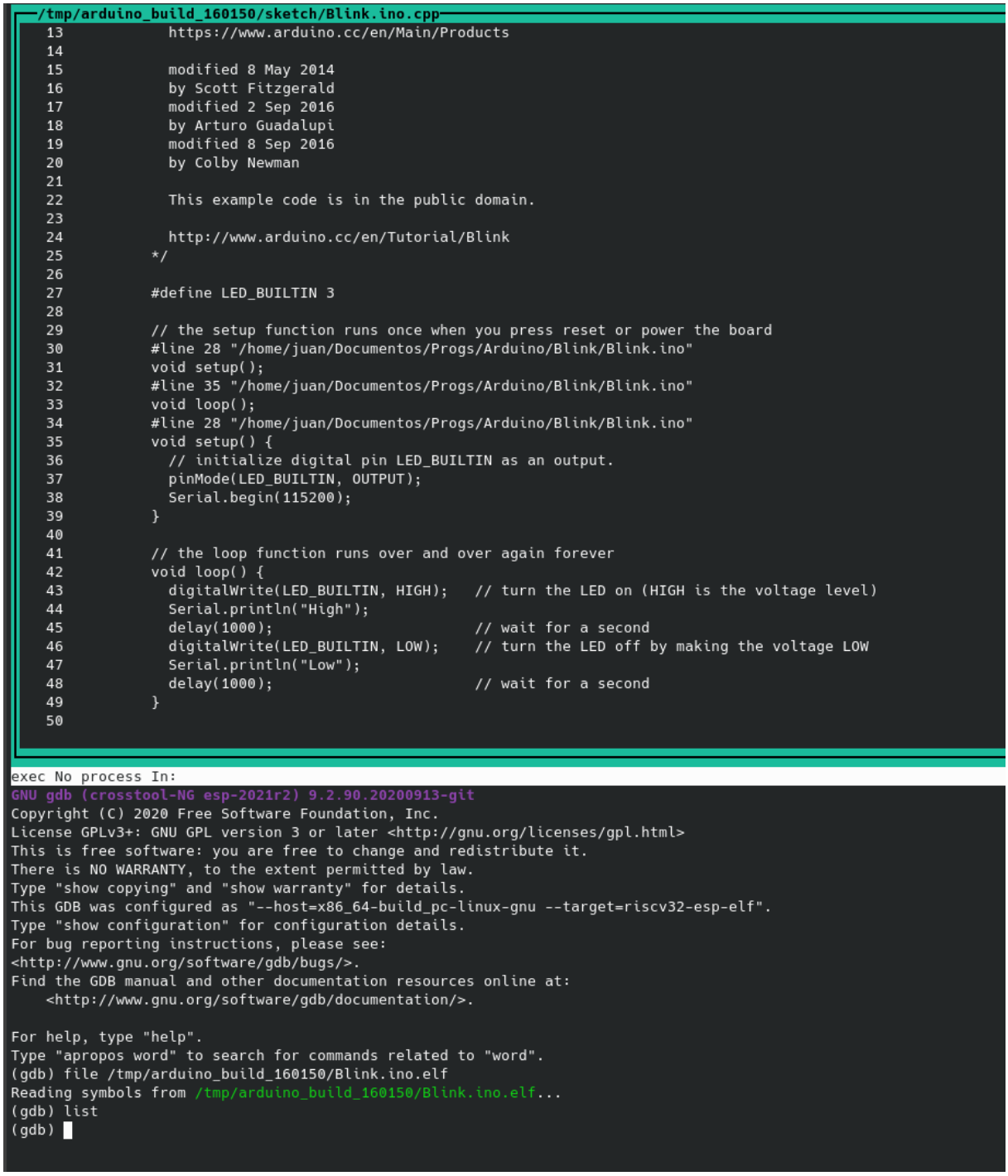

FIGURA 7: Depuración de un programa mediante GDB.