

Trabajo Final de Graduación

Bitácora de Trabajo - Entrega 4

David F. Duarte Sánchez
II Semestre 2024

1. Semana 8

1.1. Corrección de Tesis

Fecha de trabajo: 06/09/2024.

Objetivo: Corrección de observaciones realizadas a la Tesis.

Reporte de actividades
-Avances con el sistema embebido, se comienza a probar el sistema operativo a la medida que se ha elaborado para la tarjeta zedbaord, así como el funcionamiento del mismo primeramente por medio de un protocolo UART y segundo por medio del protocolo SSH, esto mediante la conexión de la tarjeta a un router para que la misma se encuentre conectada a la red.
- Se finaliza la elaboración de un manual de usuario, con el fin de hacer más fácil la evaluación del flujo de trabajo y la depuración del mismo, de esta forma se tendrán estandarizadas las versiones requeridas y las salidas esperadas en cada uno de los procedimientos que se realizan a lo largo de la experimentación.
- Se comienza la elaboración de un manual de usuario, con el fin de hacer más fácil la implementación de capas en el flujo y la depuración del mismo, de esta forma se tendrán estandarizadas los nombres de carpetas y las jerarquías en cuanto a creación de árboles de dependencias con el fin de agilizar los procedimientos que se realizan a lo largo de la experimentación.
- Implementación de una capa de prueba con el objetivo de lograr compilar cualquier archivo en el flujo de trabajo de yocto, se genera un archivo básico que imprime un hola mundo en consola en yocto, el mismo se debe de ser compilado por el flujo de trabajo con el fin de optimizar el programa para los recursos de la tarjeta objetivo.
- Solicitud de una licencia de prueba de matlab embedded coder, con el objetivo de enfocar aun mas el programa generado al hardware objetivo, Ya que de momento el programa generado por medio del simulink coder genera algunos problemas a la hora de ser compilado por gcc previo a la compilación generada por yocto

Se propone el siguiente arbol de dependencias para la capa de prueba generada

```
rex@20a26f16e5ac:~/test/poky/meta-test_control$ tree
.
├── conf
│   └── layer.conf
├── COPYING.MIT
├── README
├── recipes-core
│   └── myprogram
│       ├── files
│       │   └── main.c
│       └── myprogram_1.0.bb
├── recipes-example
│   └── example
│       └── example_0.1.bb
```

Figura 1: Estructura de la receta

```

rex@20a26f16e5ac: ~/test/poky/meta-test_control/recipes-core/myprogram/files
#include <stdio.h>

int main() {
    printf("Hello, Yocto!\n");
    return 0;
}

```

Figura 2: Programa de prueba

```

rex@20a26f16e5ac: ~/test/poky/meta-test_control/recipes-core/myprogram
DESCRIPTION = "My C program"
LICENSE = "CLOSED"
SRC_URI = "file://main.c"
S = "${WORKDIR}"

# Ensure the compilation uses the proper flags
# CC and CFLAGS are inherited from the environment
do_compile() {
    ${CC} ${CFLAGS} ${LDFLAGS} -o myprogram main.c
}

# Install the program into the correct location
do_install() {
    install -d ${D}${bindir}
    install -m 0755 myprogram ${D}${bindir}
}

# Specify additional files if necessary
FILES_${PN} += "${bindir}/myprogram"

```

Figura 3: Archivo para la compilación del sistema

```

zedboard-zynq? login: root
root@zedboard-zynq?:~# cd ..
root@zedboard-zynq?:/home# cd ..
root@zedboard-zynq?:/# cd usr/bin/
root@zedboard-zynq?:/usr/bin# myprogram
Hello, Yocto!
root@zedboard-zynq?:/usr/bin#

```

Figura 4: Sistema implementado en Zedboard