Instituto Tecnológico de Costa Rica

Profesor: Milton Villegas Lemus.

II Proyecto II Parte

Curso: Taller de Programación.

Área académica: Ingeniería en Computadores.

Realizado por:

Hansel Hampton Fallas (2019049765)

Gabriel Solano Coronado (2019033687)

Grupo 02.

II Semestre 2019.

**Introducción**

**Parte I**

El proyecto consiste en dos partes, la primera parte consiste en armar un circuito el cual fue brindado en un diagrama en las especificaciones del proyecto, además el circuito había que comunicarlo mediante un código Python versión 3.7.2 y Arduino versión 1.8.10. En este circuito se implementan varios componentes como son el NodeMCU (módulo programable) este funcionara como si fuese la placa de Arduino, el L298 (driver de motor), 74LS164 (registro de corrimiento), LM780 5(Regulador de voltaje),dos motores (Los que traía por defecto el auto de juguete) y el MPU 9250(Giroscopio), además se utilizaron leds de varios colores, diodos,. Se escribirá el código en arduino, este será escrito sobre un código brindado previamente. El código que se debe añadir al código preestablecido tiene la finalidad de implementar las funciones de movimientos, manipulación de los leds y obtención de los datos mediante las variables. Utilizando el NodeMCU se subirá el código con las instrucciones a realizar, y este se conectará mediante una red Wifi transmitiendo los datos de esta forma.

La idea principal para plantear la confección del proyecto es dividirlo, que un miembro trabaje la parte del circuito y que el otro trabaje la parte programable. Esto gracias a un acuerdo al cual llegan los dos miembros del grupo. El objetivo principal de esta parte del proyecto es realizar que realice las indicaciones enviadas por el usuario, ya sea movimientos, leds, y obtención de datos.

Completitud del proyecto: ~65%

**Parte II**

La segunda parte del proyecto consta de una interfaz mediante la cual el auto puede realizar todas las acciones que se tenían previstas, ya sea mediante el diseño de botones o de la utilización de teclas, en este caso se utilizan ambas posibilidades. Para la confección de esta interfaz se utiliza Python en su versión 3.7.2, se utilizará Tkinter en su versión 8.5 para el diseño de la interfaz.

En nuestro caso se estará trabajando también en la parte I del proyecto en conjunto con la II, ya que en la primer parte el carro funcionaba de manera correcta, mediante videos se tienen pruebas de su funcionalidad, pero en la prueba no funcionaba, por ende se dio la oportunidad de reparar esta parte del proyecto, debido a esto se están trabajando ambas partes.

El objetivo principal de esta parte del proyecto es diseñar una interfaz funcional y agradable a la vista del usuario, la cual debe contar con todas las funcionalidades especificadas previamente.

La completitud de este proyecto es de aproximadamente entre 50% y 60%

**Conclusiones:**

**Parte I**

1. Es necesario conocimiento previo sobre Arduino para el desarrollo ideal del proyecto, así como de soldadura, y una buena noción de cómo funcionan los circuitos.
2. Para efectos de orden y estética, el cable de red es una mejor opción a los jumpers.
3. Usar diagramas de datos (datasheets) es indispensable para el uso de los componentes.
4. Por cuestiones de precauciones, se dividió el circuito en dos secciones, esto para evitar complicaciones en caso de un reemplazo de emergencia.
5. Para evitar excesivos gastos económicos, se invirtió en dos baterías cuadradas de 9V, así como un respectivo cargador.
6. Se midió el voltaje periódicamente en ciertos puntos del circuito, esto para estar al tanto en caso de un fallo en el sistema y corregirlo antes de continuar.
7. Se usaron 6 baterías de 1.5V en vez de 4, con el fin de lograr un maor aguante de energía en los motores.
8. El L298 fue colocado fuera del circuito y conectado mediante jumpers, para evitar sobrecargar el espacio de componentes.
9. El circuito en funcionamiento utiliza mucha energía, lo que vuelve ineficiente la continua compra de baterías.
10. Se decidió no trasladar el circuito a placa perforada, ya que se requería de mucho tiempo del que no se disponía.
11. Usar cable UTP en vez de jumpers para la construcción del circuito sobre protoboard es más ordenado que usar jumpers.
12. Se necesita cuidado extra en cuanto al uso del cautín, ya que es un instrumento más peligroso que la mayoría.
13. Es muy importante verificar que ningún cable haga contacto con otro si no debe hacerlo, ya que puede generar cortocircuito y dañar el sistema.

**Parte II**

1. Pese a no ser la primer vez que se utiliza Tkinter se concluye que al estar diseñado para hacer interfaces agiliza mucho el trabajo a la hora de la creación de una.
2. Ya se tiene cierta experiencia con Python en este caso en su versión 3.7.2, este es un lenguaje sencillo, y tener conocimiento previo facilita el trabajo.
3. Trabajas algunos métodos con clases resulta más cómodo de implementar.
4. La utilización de los Frames resulta de mucha ayuda cuando se debe trabajar con generar tablas, en este caso la tabla debe poder organizarse.
5. La utilización de los “Tags” para distinguir entre los objetos creados con los canvas, resulta una manera sumamente sencilla de utilizar para seleccionar el objeto deseado.
6. Se utilizan distintos archivos para cada lectura diferente, debido a que es más intuitivo trabajarlos de esta forma.
7. Para circuitos que requieren una conexión inalámbrica en tiempo real, el NodeMCU presenta serias deficiencias que deben ser compensadas.
8. Las baterías cuadradas de 9V son ineficientes en circuitos de alto consumo de energía.
9. El nivel de experiencia actual nos impide realizar algunas soluciones avanzadas.
10. Las pruebas de amperaje son más eficientes para visualizar el flujo de energía.
11. La ayuda de personas con más experiencia es de suma importancia, ya que tienen más idea de cómo trabajar ciertas funciones.
12. Los cables UTP no son la mejor opción de cables ya que se quiebran con facilidad

**Recomendaciones:**

**Parte I**

1. Antes que nada, es necesario aprender a usar Arduino previamente, o como mínimo tener una noción clara de su funcionamiento, porque de lo contrario se perderá demasiado tiempo tropezando mientras se trata de aprender sobre la marcha.
2. El cable de red es una muy buena alternativa a los jumpers, ya que dan una apariencia más estética y ordenada, y resulta más fácil hacer seguimiento de las líneas de conexión. Las desventajas más marcadas es que se quiebran con relativa facilidad, y son algo difíciles de manejar sin un poco de técnica o costumbre.
3. Los diagramas de datos de los componentes son muy importantes para entender cómo funciona el interior de un circuito. Sin ellos, la creación del circuito sería a ciegas, y dificultaría exponencialmente la corrección de potenciales errores.
4. Es una buena estrategia utilizar dos protoboards unidas; es más caro que comprar una más grande, pero resulta mucho más fácil de acoplar en ciertos espacios, como el interior del carrito, además de que se ve más ordenado, y en caso de dañarse, es más fácil de reemplazar.
5. Es bueno saber interpretar los voltajes en diferentes puntos del circuito, en un buen indicador de la presencia de problemas en el circuito y facilita la búsqueda.
6. Se recomienda estar atento a la carga de las baterías, ya que algunas partes del circuito consumen más energía de la que aparentan, y podría ser la causa de un problema que no se haya ubicado.
7. Es mucho más eficiente el uso de baterías recargables, ya que el circuito activo consume energía muy rápido, lo que complica la experimentación en ausencia de repuestos a mano. Son más caras, pero es mejor que un gasto menor, pero muy continuo.
8. El utilizar operadores bitwise se recomienda, porque con estos se tiene el control de los pines, así se pueden operar de manera que cumplan con las necesidades requeridas. Así se puede elegir que led debe encender, que motor moverse etc.

**Parte II**

1. Se recomienda ir trabajando la bitácota de manera paralela a la confección del proyecto, ya que esto nos permitirá tener apuntado todo lo que se realizó y así no perder ningún detalle.
2. Se recomienda revisar el código Telemetry brindado por el asistente, ya que este nos será muy útil para poder enviar señales desde la interfaz creada por nosotros mismos.
3. Es recomendable comprender de forma clara lo que se requiere, si es así se puede trabajar de forma centrada en lo requerido.
4. Utilizar iteración o try except, para así tener un programa más fluido, ya que depende de la función la recursividad puede implicar que el programa sea lento o no funcione debido a la cantidad de llamadas recursivas que este hace.
5. Se recomienda apagar los interruptores luego de cada prueba, esto con el fin de no permitir que la batería se desgaste más de lo necesario.
6. Se recomienda realizar la documentación interna conforme se van realizando las funciones esto con finalidad de comprender y recordar que hace cada función y así mantener la buena orientación en el código.
7. Es muy recomendable utilizar el estado HIDDEN en las imágenes para así poder generar transiciones, o movimientos de imágenes.
8. Se recomienda utilizar el .bind() para el reconocimiento de teclas, ya que si solo se utilizan botones en pantalla, esta se saturará de objetos, por ende no se verá estéticamente bien.
9. Se recomienda dar nombres significativos a las variables y funciones, ya que al ser un programa extenso pueden llegar a generar confusión algunos nombres, o inclusive escribirlos mal lo cual nos va a generar errores.
10. Se recomienda utilizar print() en algunas partes del código para saber s está funcionando como debería, también para conocer si el valor de algunas variables es el deseado.
11. Luego de realizar pruebas con print () es recomendable quitarlo ya que en algunos casos muy específicos puede generar lentitud en el programa.
12. El NodeMCU es un integrado bastante práctico a la hora de implementarlo en un circuito, especialmente sobre protoboard, dado su tamaño y la inclusión de un módulo WI-FI (ESP8266), pero su consumo energético es muy elevado, característica que debe ser compensada si el circuito requiere de una conexión en tiempo real.
13. Una buena forma de detectar fugas en el voltaje es desconectar individualmente la alimentación de cada componente que reciba voltaje de la misma fuente. Si el componente en cuestión está consumiendo más energía de lo normal, el flujo de voltaje debería regularse al desconectarlo.
14. Al realizar investigaciones en internet en busca de soluciones a un problema, se suelen encontrar algunas que, si bien pueden resultar efectivas, conllevan mucho riesgo para ejecutarlas con nuestro nivel de experiencia, así que es recomendable fijarse muy bien en la viabilidad de las soluciones encontradas.
15. Se recomienda utilizar Tkinter para diseñar la interfaz, ya que esta cuenta con muchas facilidades, botones, labels, canvas, entry. En general es muy intuitiva-
16. Es de suma importancia manera bien el tiempo, dedicarle tiempo tanto a proyectos como a otros cursos, para así no tener que dedicar todo el tiempo a una u otra tarea y no tener problemas con tener que dejar algunas funcionalidades sin cumplir.
17. Se recomienda utilizar las teclas WADS para movimiento del carro, ya que estas son las que generalmente se utilizan en videojuegos y estas resultan muy cómodas al usuario.

**Análisis de resultados:**

**Parte I**

El proyecto resultó bastante completo, en general; la protoboard base quedó bastante ordenada, los cables recortados a medida en su mayoría, entre otros detalles que contribuyeron al orden y la estética. En cuanto al funcionamiento, los comandos básicos también fueron exitosos, así como los de luces; sin embargo, los movimientos especiales se quedan muy deficientes, o hasta incompletos.

A raíz de un proyecto previo similar a éste, iniciamos un poco confundidos por tratar de adaptar nuestros conocimientos al nuevo circuito, que funciona de forma bastante diferente, por lo que padecimos varios problemas entre los que destaca uno: el L298 es un driver que administra a cuál motor le llega energía, y cuánta. Para enviarle señales, usualmente lo conectábamos al NodeMCU directamente, pero en esta ocasión debía conectarse al registro de corrimiento, que recibía todas las señales enviadas y las repartía, y no sabíamos cómo manejarlo de esa forma. Preguntando a diversas personas nos encontramos con los bitwise operators, usados para operar secuencias binarias de bits, con los que logramos enviar las instrucciones correctas a través del registro.

Otro problema recalcable estuvo en la conexión del L298 con la protoboard, de la cual, por cuestiones de espacio, lo sacamos a través de jumpers sueltos que pegamos entre sí con pegamento. Posteriormente, al ordenarle al carrito que acelerara, lo hacía de forma intermitente y durante un intervalo de tiempo aleatorio. Nos dimos cuenta poco después de que ese periodo se alargaba colocando en driver de cierta forma, por lo que sospechamos de los cables; al estar pegados entre sí, fue necesario cambiarlos todos, pero con eso arreglamos el problema.

También nos topamos con fuertes ineficiencias de energía en la batería de 9V, lo que nos obligaba a comprar continuamente más baterías y a reemplazarlas antes de que se descargaran por completo. Nuestra solución fue comprar dos baterías recargables; de esta forma, podríamos cargar una mientras usábamos la otra, y no evitábamos el innecesario gasto de dinero en tantas baterías normales. A las baterías de 1.5V también le añadimos 2 baterías, para darle mayor aguante a los motores.

Sin embargo, nuestro principal problema radicó en la ausencia inicial de multímetro y cautín, este último no conseguido hasta los últimos días; esto nos retrasó mucho debido a la imposibilidad de realizar pruebas de voltaje y continuidad, por lo que no sabíamos qué buscar cuando el circuito no funcionaba; a pesar de esto, nos recuperamos bastante bien, y llevamos el proyecto bastante lejos.

**Parte II**

-Se logra la implementación de los métodos de ordenamiento así como la obtención de los datos de un archivo de texto.

-Se diseña una interfaz amigable y agradable para el usuario.

Se logra el envío de señales al NodeMCU parcialmente, esto debido a que no teníamos manera e probar si el programa funcionaba de manera correcta, esto debido a problemas presentados por el carro a cercanía de la entrega del segundo proyecto. Por ende no se podía probar si el código funcionaba así que se intentó programar sin parte de hardware, lo cual no resulta de buena manera ya que no se observa el comportamiento real del carro. Un problema que se tuvo fue que las señales se enviaban y respondía time out como era de esperarse, pero en un momento dado al apagar y encender el laptop donde se estaba programando y dejó de responder con esta señal. Se intentó modificar y realizar cambios pero no funcionaba como debía.

El problema presentado anteriormente era con respecto a cuándo responde a las señales enviadas de los movimientos.

Las funciones que tienen mayor relación con el MPU no fueron implementadas, yaw,pitch, roll, ya se obtienen los valores mediante el código de arduino, pero estas funciones tampoco pudieron ser probadas, y al no poder probarlas resulta muy compleja la programación sin poder realizar las pruebas conforme es requerido.

**Imagen que contiene texto, mapa

Descripción generada automáticamente**

**Plan de pruebas**

**Parte I**

-Se realizarán pruebas de voltaje y continuidad a lo largo de la construcción del circuito, con el fin de prevenir errores en la construcción del mismo.

-Se realizará una prueba general en el circuito, en busca de fallas de voltaje o conexiones defectuosas que puedan causar un mal funcionamiento.

-Se verificará que el programa de Arduino compile correctamente antes de intentar avanzar con las pruebas, así como que las operaciones y funciones lógicas estén bien interpretadas para lo que se espera que el carro haga.

-Se verificará el funcionamiento de las luces, y los diferentes patrones que deben seguir; repitiendo pasos anteriores, se corregirán los errores que se presenten.

-Se probará el movimiento básico del carro: adelante y atrás, izquierda y derecha, en ese orden; primero comprobando el funcionamiento en el aire, luego sobre una superficie sólida. Repitiendo pasos anteriores, se corregirán los errores que se presenten.

-Se realizarán pruebas para los movimientos de patrones básicos del carro, como el circular, el infinito y el especial, en ese orden; primero comprobando el funcionamiento en el aire, luego sobre una superficie sólida. Repitiendo pasos anteriores, se corregirán los errores que se presenten.

-Se comprobará el funcionamiento de los movimientos especiales del carro que involucran un sensor, tales como el posicionamiento al norte, sin orden específico, directamente sobre una superficie sólida. Repitiendo pasos anteriores, se corregirán los errores que se presenten.

-Se repetirán pruebas generales para el funcionamiento completo del auto, así como la carga de las baterías, para verificar que todo el auto esté en funcionamiento.

**Parte II**

-Se realizarán pruebas de continuidad para determinar si hay algún cable mal conectado. Si se encuentran errores en el circuito se corregirán, sino se encuentran se procederá a indagar algún otro problema que pueda presentar, ya sea de circuito o de algún elemento/módulo.

-Si se corrigen los problemas de conexión del NodeMCU se procederá a programar la parte del MPU, la cual no fue implementada en la primer parte, por temas de errores que se presentaron al conectarlo e intentar implementarlo.

-Se creará un archivo txt para probar la lectura de texto y ordenamiento, si funciona esta parte se dará por concluida

-Se prepará un diseño de interfaz el cual será mostrado al otro miembro del grupo, para recibir críticas constructivas, al tener la interfaz amigable y que se vea bien estéticamente se dará por concluido el diseño.

-Al implementar la conexión con el NodeMCU se realizará la prueba de todos los movimientos, para determinar si la parte de la programación está correcta, si no funciona se corregirán los problemas y si es correcta se pasará a otras partes de la programación.

-Se intentarán implementar las clases, fue previamente consultado si estás podían ser utilizadas con la confección del proyecto, si no se pueden implementar se buscará otro método para realizar la parte que se está implementando.

-Se utilizará el método de ordenamiento. Conocido como bubble sort, si este resulta de dejará con este, si no funciona de manera eficiente se buscará una alternativa.

**Bitácora:**

**Viernes 18/10/19:** Hansel realizó las primeras compras de componentes para la construcción del circuito. Algunos, como el driver L298, fueron reciclados de proyectos anteriores, lo que redujo el gasto.

**Sábado 26/10/19:** Con varios días de retraso, se inicia la construcción del circuito. Para ese momento carecíamos de algunas herramientas como el cautín y un multímetro, lo que dificultó enormemente la labor y casi imposibilitó algunas tareas, como la integración de las baterías al circuito; eso se resolvió temporalmente con ayuda de un cautín dañado al que había que aplicar presión sobre la punta para que calentara a un nivel utilizable, y su punta no era la más apta para una soldadura delicada, por lo que las conexiones no eran muy estables. Se estableció que Hansel fuera el principal encargado de la parte física que involucraba el circuito, mientras que Gabriel se enfocara más en la parte programada.

**Lunes 28/10/19:** Gabriel compró resistencias nuevas, ya que las que disponíamos no estaban en muy buen estado, y las reemplazamos; también una batería de 9V con su respectivo cargador, por motivos de ahorro. Hansel ya tenía el circuito cuidadosamente terminado, pero sin multímetro, no se podía hacer mucho más.

**Miércoles 30/10/19:** Hansel pidió prestados en la asociación de estudiantes de CE un cautín y un multímetro, para poder probar y corregir el circuito, que para entonces ya teníamos construido, aunque con muchos errores algo básicos debido a la incapacidad para probarlo; sin embargo, el cautín estaba dañado, así que seguimos varados en ese aspecto.

**Jueves 31/10/19:** Hansel trató de cambiar el cautín en la asociación, pero ya se habían agotado, y esa semana estábamos cortos de dinero para comprar uno, así que tratamos de conseguir uno prestado por nuestra cuenta. Gabriel fue a comprar diodos nuevos de repuesto, pues los que teníamos estaban dañados.

**Sábado 02/11/19:**A pesar de los contratiempos, Gabriel empezó a desarrollar el programa de comandos. Tuvimos un inconveniente con el método de emisión de órdenes con los motores, que estaba incorrecto porque las entradas del L298 que conectaban con los motores conectaban con el registro en vez del NodeMCU y no sabíamos cómo manejarlo; invertimos una gran parte del día investigando el método correcto.

**Domingo 03/11/19:**Encontramos el método de los bitwise operators, operadores lógicos a nivel de bits, con los que logramos que el motor delantero funcionara, pero sólo de forma intermitente, lo que era muy raro.

**Lunes 04/11/19:**A base de pruebas descubrimos que el motor reaccionaba mejor al mover los cables del L298, lo que nos hizo pensar que estaban de alguna forma interiormente en mal estado.

**Martes 05/11/19:**Gabriel compró los nuevos jumpers del L298, y el problema se corrigió cuando Hansel los corrigió. Gabriel terminó de configurar los movimientos por defecto de los dos y preparó el programa para armar los movimientos complejos. También compró un cautín, con el que Hansel logró corregir la soldadura de los interruptores de las baterías.

**Jueves 07/11/19:** Gabriel trató de elaborar los movimientos especiales mientras Hansel rellenaba los espacios vacíos de la documentación.

**Viernes 08/11/19:**Luego de instalar el MPU soldado, el programa aparentemente se se desconfiguró: los motores dejaron de responder y hubo una fuerte deficiencia de energía por parte de la batería de 9V; al desconectarlo, se arregló a medias, pues los motores dejaron de hacer lo que debían hacer.

**Sábado 09/11/19:**Se revisa el circuito debido al problema que causaba con el movimiento nulo de los motores, pero al moverse antes no era problema de programación, lo que nos llevaba a tener un problema en el circuto, Hansel se encargó de corregirlo.

**Domingo 10/11/19:**Se intentó solventar el problema de los motores ya con el ciruito corregido, pero teníamos un problema y era que la batería de 9 voltios se descargaba de manera muy rápida lo cual no permitía hacer pruebas de la mejor manera. Por ende no se logró nada este día.

**Lunes 11/11/19:**Día de entrega del segundo proyecto; ambos optamos por cambiar todas las baterías de 1.5 voltios, lo cual solucionó el problema de la descarga de la batería de 9 voltios, por ende se realizaron las pruebas necesarias, en efectivo el cambio que aplicó Hansel en el circuito resultó correcto.

**Martes 19/11/19:** Día de la defensa del segundo proyecto, durante la cual se sufrió las consecuencias de insuficiencia energética, ya que la batería se descargaba muy rápido, y el voltaje que emite es de aproximadamente la mitad de lo que debería. Hansel hizo algunas pruebas en el tiempo extra que fue otorgado, desconectando la alimentación de 5V de los diferentes módulos del circuito, y así se descubrió que, al desconectar el NodeMCU de su alimentación de 5V, el voltaje en todo el circuito se regulaba; al intercambiarlo con el NodeMCU de un compañero, y al ver que el problema persistía, se dedujo que el Node dañado no era la causa. No se logró corregir para esta ocasión, debe corregirse para la entrega del tercer proyecto.

**Miércoles 20/11/19:** Este día se creó el plan por parte del líder, se conversó entre ambos miembros del grupo con la repartición de tareas que cada uno debe llevar a cabo, al no existir objeciones se lleva a cabo dicho plan. Gabriel inicia con el método Poyla de lo solicitado en el proyecto, además de buscar algunas imágenes las cuales serán utilizadas en la interfaz del proyecto. Este día el internet de nuestra compañía estuvo suspendido a nivel nacional, por tanto no se pudo trabajar de manera fluida

**Jueves 21/11/19:** Gabriel Inicia con la parte de la programación, se reutilizó código de la programación del primer proyecto, la cual consistía en transición de ventanas, lo cual aceleró el trabajo. A demás va trabajando con la documentación de manera paralela a la programación. Se implementó la parte de ordenamiento mediante el uso de clases, lo cual implicó un poco de investigación para así lograr implementarlo en el código. Hansel realizó algunas pruebas de circuito más, pero el problema persiste. Además, al encenderlo, todo el voltaje se reparte casi instantáneamente, y se reparte de forma uniforme por todo el circuito, entonces se dificulta encontrar anomalías.

**Viernes 22/11/19:** Gabriel continúa con la parte del código, además se trabaja más la parte de la documentación, asimismo Gabriel va generando el diseño de la interfaz, ya con las imágenes que van a ser utilizadas y se le pregunta a Hansel, el otro miembro del grupo que opina acerca del diseño, el cual da algunas recomendaciones las cuales Gabriel anota para realizarlas más adelante. A Hansel se le ocurre una idea: si la fuga se manifiesta en la alimentación de 5V del NodeMCU, pero éste no estaba dañado, quizá el problema estaba en las conexiones que derivaban del mismo; desconectamos algunas conexiones del NodeMCU, pero no se observó la normalización de voltaje anterior, lo que hace pensar que el problema es interno del NodeMCU.

**Sábado 23/1/19:** Gabriel intenta implementar la conexión entre el NodeMCU y la interfaz, además se trabaja en la documentación, adjunto a esto se agrega la documentación interna al código que se lleva actualmente. A base de algunas investigaciones nos enteramos de que es bastante común en sistemas que usan el NodeMCU y que requieren de un funcionamiento constante en tiempo real, así que es un problema, pero no un error. Una de las soluciones encontradas consiste básicamente en desmontar el regulador interno del NodeMCU y desconectar uno de los pines del esp8266, que son los responsables aparentes de reducir la energía de alimentación, pero a nuestro parecer es muy arriesgada; la única solución alterna a nuestro alcance es alimentar el circuito con una fuente más potente, ya que las baterías de 9V poseen muy poca corriente. Hansel se reunió con el asistente del curso por la tarde, en busca de ayuda para diagnosticar el problema de energía; finalmente, a raíz de una serie de comportamientos anómalos, se llega a la convención de rearmar el circuito desde cero. Sin embargo, las anomalías se repiten incluso cuando sólo está conectado el NodeMCU y el regulador junto a la fuente.

**Domingo 24/11/19:** Gabriel continúa con la programación, la cual para este punto ya enviaba los comandos y como no teníamos la parte del hardware o podíamos probar, cuando uno enviaba los comandos daba como respuesta “time out”, pero en un punto de la tarde se reinició la laptop donde se estaba programando el código y dejó de funcionar el envío de señales, se trató de arreglar este problema, Se termina con la documentación externa. No se logra solucionar el problema.

**Lunes 24/11/19:** Ambos miembros dan unas últimas revisiones a la documentación para arreglar los últimos detalles y así entregar el proyecto.

**Bibliografía**

• Desconocido. (s.f). Guía de referencia de Arduino. Recuperado de: https://www.arduino.cc/reference/es/

• Desconocido. (s.f). Stack Overflow – All Questions. Recuperado de: https://stackoverflow.com/questions

• Lundh, Fredrik (s.f). Zone.Effbot. Recuperado de: http://effbot.org/zone/index.htm

• Código brindado por José Morales.