

Proyecto Final de Curso

Implementación de un sistema mecatrónico basado en un microprocesadores y microcontroladores.

I. Objetivo General.

Familiarizar al estudiante en la utilización de tarjetas basadas en microprocesadores y microcontroladores de manera que se aprovechen todos los recursos de software y hardware mediante la construcción de un sistema mecatrónico a la medida.

II. Descripción del trabajo.

En este proyecto de diseño en ingeniería el grupo deberá enfrentarse una labor de complejidad media para estudiantes de ingeniería de séptimo semestre, por ello este trabajo requiere (al igual que todo proyecto) de una adecuada planificación de las tareas a realizar por parte de los estudiantes. La evaluación se compone una serie de avances de carácter formativo como el objetivo de guiar y retroalimentar los estudiantes en el proceso de diseño, y dos evaluaciones sumativas que se detallan en secciones posteriores.

En este proyecto se requiere implementar un sistema mecatrónico que esté formado por una torreta y varios sistemas de persecución de objetivos. En términos generales la torreta cuenta con las siguientes características:

- Capaz de apuntar en un volumen de media esfera
- Puede tener más de un cañón según lo que el equipo de trabajo determine necesario
- Debe de ser capaz de disparar a un objetivo hasta 20cm de distancia. Estos 20cm se toman a partir de la punta de cañón.
- Disparar no se limita a enviar objetos físicos sino a marcar con una señal laser o similares.
- Los objetivos a disparar son de los siguientes colores: Rojo, verde, azul, negro. Se recomienda buscar objetivos saturados.
- Al final de cualquier sistema de persecución la torreta regresa a una posición de reposo definida por los estudiantes.
- En ningún momento el usuario debe de tener que interactuar con la torreta, el usuario solo controla la torreta mediante un control externo en los modos manuales y la interfaz. La única excepción a esta regla es para la **recarga** de balas, dardos, etc.

La operación de la torreta se basa en 3 modos de uso distintos, todos accesibles mediante una interfaz controlada por un microprocesador, los modos no pueden interrumpirse entre si y deben de estar disponibles en todo momento al usuario; en otras palabras, no se aceptan modificaciones de la planta en medio de la evaluación para presentar cada uno de los modos. **La complejidad de la interfaz queda a discreción del equipo de trabajo.**

A continuación, se detalla cada modo de uso:

Modo Manual o Modo Juego de Puntería: Se basa en proveerle a un usuario del control total de la torreta para que participe en un juego de puntería.

- El área de juego consta de una pared sobre la cual se colocan distintos objetivos de colores. La pared tiene dimensiones 25x25cm donde cada objetivo es un cuadrado 5x5cm.
- La interfaz debe indicarle al usuario de forma aleatoria cual objetivo debe buscar y disparar
- Si el usuario atina debidamente se suman dos puntos, si falla se le resta un punto.
- El juego tiene una duración de 2 minutos, la interfaz debe de indicar debidamente el tiempo restante de juego.
- El sistema debe de mantener una lista de “mejores puntajes” con un máximo de cinco puntuaciones, esta lista debe de perdurar aún si el sistema se apaga completamente y debe de actualizarse debidamente según el resultado de cada juego.
- La interfaz debe de proveer una imagen desde el punto de vista de la torreta para que el usuario apunte debidamente.

Modo Programable: En este modo la torreta lee un archivo .txt con una lista de instrucciones y las ejecuta 1 a 1. Debe de asegurar la siguiente funcionalidad:

- Moverse en un eje/rotación/etc X cantidad de unidades.
- Movimiento simultaneo en dos o más ejes/rotaciones en X,Y unidades.
- Pausa de X segundos.
- Regresar a posición de reposo.
- Disparo de un cañón (En caso de tener múltiples cañones debe de poder elegirse cual cañón).
- Apuntar de forma directa vertical.
- Movimiento a posición cardinal sobre la base de la media esfera.

Modo Watchdog: En este modo la torreta empieza apuntando a su objetivo según color y lo persigue en tiempo real según este se mueva.

- Si el objetivo desaparece la torreta se detiene en su lugar hasta que lo vuelva a observar.
- El usuario puede detener este modo desde la interfaz.
- La interfaz debe mostrar en tiempo real el estado del objetivo: En mira, persiguiendo, perdido.
- **Grupos de 5 Estudiantes:** Cuando se detiene el modo watchdog se debe de generar un archivo txt con instrucciones para que el Modo Programable replique el comportamiento del watchdog.

El grupo deberá utilizar un microprocesador, particularmente una Raspberry Pi 2/3, para el procesamiento de información, manejo de interfaz, etc. Por su parte, un sistema basado en un microcontrolador se encargará del control de la torreta y todos los sistemas de medición y control que no sean de captura de imágenes.

A continuación, se describen los requerimientos generales que debe de cumplir el microprocesador:

- La unidad central de procesamiento debe ser una tarjeta de desarrollo Raspberry pi 2 / Raspberry pi 3.
- Puede manejar dispositivos de captura de imágenes si el grupo lo define necesario.
- El software debe ser desarrollado en lenguaje C/C++ o Python **únicamente**. No se permitirá una mezcla de los dos.

- **Grupos de 5 Estudiantes:** La comunicación con el microcontrolador se realizará mediante protocolo RS-232, I2C o SPI.
- El microprocesador maneja todos los algoritmos de análisis de imágenes, interpretación de texto para el modo programado, manejo de interfaz, entre otros.
- Debe de utilizar herramientas de paralelismo para el manejo de sus labores.

A continuación, se describen los requerimientos que deberá cumplir el sistema basado en microcontrolador:

- El microcontrolador deberá ser seleccionado por los estudiantes y deberá ser a la medida.
- El mismo (modelo) microcontrolador podrá ser utilizado por un máximo de 2 grupos.
- No se permite el uso de microcontroladores de la familia ATmega.
- El microcontrolador se encargará del movimiento de la torreta.
- El microcontrolador se encarga de gobernar el sensor o sensores que permitan la correcta operación de la torreta.
- El microcontrolador se encarga de gobernar el actuador o actuadores que permitan la correcta operación de la torreta.
- **Grupos de 5 Estudiantes:** El microcontrolador deberá comunicarse con el sistema de identificación únicamente por protocolo de comunicación RS-232, I2C o SPI.
- Inicialmente, el sistema se encontrará inactivo. Para que la torreta comience a funcionar será necesario presionar un botón de inicio.
- El uso de interrupciones para el microcontrolador es de carácter obligatorio, no se acepta polling en el código de la solución.
- El sistema deberá ser capaz de operar de acuerdo con las condiciones especificadas y siguiendo todas las restricciones de diseño, debe de retornar a una posición de reposo al finalizar cada modo. De no cumplir con lo anterior, el proyecto se encontrará dentro de la categoría: "No Funcional".
- Si el proyecto se encontrara dentro de la categoría No Funcional, la calificación se realizará tomando como base los 70 puntos y aplicando la rúbrica de evaluación abajo mostrada.

III. Consideraciones generales:

1. El proyecto debe realizarse en grupos de 5 personas.
2. **Todos** los miembros del grupo deberán estar en la presentación final de lo contrario se le asignará un 0 al estudiante que falte y no presente la justificación respectiva.
3. Se deberá entregar un informe final con todo el diseño detallado, así como la evaluación de las diferentes soluciones al problema, análisis de estas e implementación del proyecto. **Máximo 6** páginas formato IEEE doble columna. Todos los grupos de trabajo deben llevar una bitácora donde cuentan sus vivencias, toma de decisiones, anotaciones de reuniones de los grupos, cálculos, bosquejos necesarios y todo aquello que considere pertinente para un adecuado seguimiento del proceso de diseño.

IV. Entregables:

Nota: Los avances están ligados a una semana. Se asume que el avance se entrega el día de clases de cada semana respectiva.

Avance 1: Se deberá presentar en Semana 3 al final de la clase, un reporte donde se presente una propuesta de diseño de software para el modo programado y watchdog. El reporte deberá contener en prosa y con el apoyo de diagramas de flujo una descripción detallada del algoritmo y la técnica a utilizar. Deberá contener un diagrama de flujo y una investigación de herramientas para paralelizar el software en C o Python que permitan maximizar el rendimiento del trabajo.

Avance 2: Se deberá presentar en Semana 6 un informe de avance, donde se muestre los criterios utilizados para la selección del microcontrolador que utilizarán en el proyecto, deberá redactar un documento de a lo sumo 1 página donde se comparen cualitativamente como mínimo 3 microcontroladores y demuestren por qué el microcontrolador escogido es el que más se adapta a la aplicación (es recomendable el uso de tablas). Para eso compare los recursos que los candidatos poseen, cantidad de unidades funcionales, potencia que consumen, entradas, salidas, costo, cómo se programan, etc. Verifique que existe un programador o que este es de fácil acceso para evitar tener problemas con la programabilidad de este.

Avance 3: Se deberá presentar en Semana 10 una descripción del funcionamiento del sistema basado en microcontrolador, diagramas de flujo del software, diagrama de bloques de hardware, esquemático, construcción mecánica del sistema y todas las consideraciones de diseño utilizadas. No olviden que este es un paso fundamental en su diseño y no se permite el cambio del microcontrolador una vez este le fue asignado.

Avance 4: Se deberá presentar en Semana 14 una muestra funcional del algoritmo del modo de programación. A su vez una demostración del algoritmo o forma de control del cañón en el modo manual.

La entrega final del proyecto se realizará en Semana 18, mientras que el informe final deberá subirse en formato pdf al TecDigital antes de las 11:59 p.m. del mismo día, no se admiten documentos fuera de plazo o que no cumplan con la forma de entrega.

V. Evaluación

Rubro	Puntaje	Fecha de entrega
Avance 1: Informe de diseño de software de modo programación y watchdog.	Formativo	Semana 3 Día de Clases
Avance 2: Informe de selección y diseño, microcontrolador.	Formativo	Semana 6 Día de Clases
Avance 3: Informe del funcionamiento del microcontrolador	Formativo	Semana 10 Día de Clases
Avance 4: Presentación del algoritmo de programación y control manual	Formativo	Semana 14 Día de Clases
Entrega Final	70 pts	Semana 18
Informe Final	30 pts	Semana 18
Nota Final	100	

Evaluación de la Entrega final del Proyecto

Criterio	Detalle	Valor
Uso de interrupciones y Paralelismo	El uso de interrupciones está presente en la solución del microcontrolador. El uso de paralelismo está presenta de forma significativa en la solución del microprocesador. El software es capaz de evidenciar su uso.	Excelente (4pts): Posee paralelismo significativo e interrupciones Bueno (2pts): Posee solo paralelismo o solo interrupciones Malo (0 pts): No posee ninguno de los dos.
Movimiento de torreta controlado por microcontrolador	El microcontrolador es capaz de controlar la cantidad de motores que sean necesarios para poder desplazar el sistema en todos sus ejes y cubrir el volumen de media esfera.	Excelente (5pts): El sistema controla sus ejes de tal forma que abarca un volumen de media esfera Bueno (2pts): El sistema posee problemas de movimiento. Esto se puede traducir en: Problemas en abarcar al menos un cuarto de esfera, problemas en abarcar las partes superiores de la esfera, entre otros Mal (0pts): El sistema posee problemas graves de movimiento. Solo es capaz de pequeños arcos de rotación o ninguno.
Comunicación Microprocesador-Microcontrolador	La comunicación entre el microprocesador y el microcontrolador es del tipo RS-232, I2C o SPI. No existe ningún otro medio de comunicación entre ambos sistemas y no utiliza más de un método.	Criterio binario (5pts): Funciona o no funciona.
Grupos de 5 Estudiantes		
Posición de Reposo	Después de cualquier modo de operación, y en el momento de encender el dispositivo, la torreta busca una posición de reposo.	Excelente: Cumple el requisito correctamente (6pts) Bueno: Cumple el requisito en 2 de los 3 modos de operación (4pts) Mínimo: Cumple el requisito en 1 de los 3 modos de operación (2pts) Mal: Para todos los modos la posición de reposo es manual (0pts)

Modo Manual: Control total de la torreta	El sistema permite un control simultáneo de todos los ejes y elementos para controlar totalmente la torreta. Esto implica la existencia de algún mando de control unido al microcontrolador.	Excelente (7pts): El requisito se cumple totalmente. Bueno (4pts): El control manual funciona pero no puede operar simultáneamente los ejes de la solución. Mínimo (2 pts): El control manual no permite abarcar todo el volumen de la esfera Mal (0pts): No existe control manual. O solo permite disparar.
Modo Manual: Detección de aciertos	El sistema es capaz de detectar correctamente el acierto o fallo de un disparo	Excelente: El sistema pasa 10 pruebas correctamente (8pts) Bueno: El sistema pasa entre 7 y 9 pruebas correctamente (5pts) Mínimo: El sistema pasa entre 3 y 6 pruebas correctamente (3pts) Mal: El sistema pasa de 0 a 2 pruebas correctamente (0pts)
Modo Manual: Puntuación e interfaz	El sistema utiliza la interfaz en el modo de juego como es solicitado. A su vez maneja adecuadamente las puntuaciones.	Excelente: El sistema cumple el requisito correctamente (5pts) Bueno: El sistema falla en una de las necesidades de la interfaz de usuario o puntuaciones (2pts) Mal: El sistema falla dos o más de las necesidades de la interfaz (0pts)
Modo Manual: Tiempo de Juego	El sistema respeta el tiempo de juego estipulado en las instrucciones.	Criterio binario, funciona o no funciona (1pt)
Modo Watchdog: Persecución en tiempo real	El sistema logra seguir al objetivo en lo que este se mueve. Se asume que el objetivo empieza en una posición conveniente para la posición de reposo de la torreta.	Excelente: El requisito se cumple correctamente (10 pts) Bueno: El requisito se cumple, pero la persecución empieza hasta 2 segundos después de que se mueve el objetivo. (7pts) Mínimo: El objetivo se cumple pero la persecución empieza hasta 10 segundos después de que se mueve el objetivo (3pts). Mal: No hay persecución (0pts)

Modo Watchdog: Replicación de secuencia	Al finalizar el modo watchdog el sistema debe generar un archivo .txt que permita replicar los movimientos realizados durante la persecución, debe de considerarse los tiempos de espera donde no hubo movimientos. Grupos de 5 estudiantes	Excelente: El txt permite una replicación casi exacta de las labores del watchdog (8pts) Bueno: El txt realiza una replicación a groso modo dando prioridad posiciones finales (4pts). Mal: No hay txt para replicar los movimientos o el txt generado no se puede usar por el Modo Programable
Modo Programable: Operaciones	El sistema puede utilizar archivos txt con sets de instrucciones para controlar el movimiento de la torreta.	Se otorga 1 punto por cada tipo de instrucción que pueda ejecutar correctamente. (Max=8pts)
Modo Programable: Instrucciones no soportadas	El sistema debe de devolver un error en interfaz si se encuentra con una instrucción que no es capaz de interpretar.	Criterio binario funciona o no funciona (5pts)
Interfaz	El sistema presenta una interfaz que permite acceder a todos los modos, a su vez la interfaz permite al usuario ingresar la dirección del txt a utilizar en el modo programable y el color del objetivo a seguir/buscar en el modo watchdog.	Criterio binario funciona o no funciona (3pts)
Bitácora	Todos los grupos de trabajo deben llevar una bitácora donde cuentan sus vivencias, toma de decisiones, anotaciones de reuniones de los grupos, cálculos, bosquejos necesarios y todo aquello que considere pertinente para un adecuado seguimiento del proceso de diseño.	Criterio binario (4pts): Posee o no posee bitácora legible y completa según requerimientos.
Extra: Modo "Repetición de partida"	El sistema es capaz de "grabar" el último juego realizado por el usuario y da la opción en la interfaz principal de ver una repetición de este. Debe de contemplar tiempos en los que el usuario no hizo nada y los disparos.	Excelente: Se cumple el requisito totalmente (15pts) Mínimo: Se cumple excepto por los momentos de reposo del usuario (7pts) Mal: No presenta el modo o tiene más defectos que los dados por el rubro "Mínimo"

Nota 1: No olviden acatar todas las instrucciones que se encuentran en el programa del curso.

Nota 2: Todos los avances formativos se presentarán al profesor durante la clase del día de la entrega para que los grupos reciban retroalimentación en su proceso de diseño, además será necesario presentar la bitácora en cada uno de los avances.

Nota 3: Tenga en cuenta que el tema de interrupciones será cubierto en las últimas semanas del curso, se aconseja discutir con su profesor la implementación propuesta.

Nota 4: En caso de que durante la evaluación se de con un error que no calce exactamente con la rúbrica el profesor tiene la capacidad de asignar una penalización adecuada a su criterio. Es deber del profesor documentar esta nueva penalización para lograr una justa evaluación de todos los equipos.

Nota 5: La rúbrica de evaluación del informe se encuentra disponible en el TecDigital.

*El profesor se reserva el derecho a admitir un cambio de microcontrolador en casos excepcionales, mediante un documento escrito donde el grupo exponga la razón o razones de peso que justifican el cambio, una nueva valoración exhaustiva del microcontrolador y un cuadro comparativo entre todas las opciones estudiadas. Además, será necesario presenta un esquemático completo de las conexiones y todos los diagramas de flujo asociados con el código a implementar en el microcontrolador.