

**ISAE UNIVERSIDAD**

**FACULTAD DE CIENCIAS TECNOLÓGICAS**

**Facultad de Ciencias Tecnológicas**

**Licenciatura en Informática con Énfasis en Sistema de Información**

**Seminario de Robótica Fischertechnik- Programación**

**Guía Didáctica**

**Profesora del curso**

 **Jhinezhka Vergara**



**“El Futuro está en tus Manos”**

**AUTORIDADES**

**Dra. Xiomara de Arrocha**

**Rectora**

**Mgter. Ivonne Arrocha**

**Vicerrectora**

**Profesor René Atencio**

**Director Académico**

**Ing. Constantino Guerra**

**Secretario General**

**Licda. Teresa Perea**

**Directora de Administración y Finanzas**

**Robótica Fischertechnik- Programación**

**DESCRIPCIÓN**

Con fischertechnik se logra el desarrollo intelectual en programación, el lenguaje gráfico utilizado para fischertechnik es ROBO PRO, el cual proporcionará todos los elementos clave de un lenguaje de programación moderno, como las matrices, las funciones recursivas, objetos, eventos asíncronos y procesamiento paralelo. Esto lo convierte en una herramienta útil incluso para programadores profesionales. Los programas se traducen directamente en lenguaje máquina para la ejecución de máxima eficacia. Incluso los programadores avanzados encontrarán que ROBO PRO no conoce límites.

**OBJETIVO GENERAL**

* Construir y programar de máquinas capaces de realizar tareas con flexibilidad, robustez y que sean eficientes para el trabajo que fueron creados, al igual que estimular el acercamiento personalizado a la robótica, logrando la adaptación de los estudiantes en los procesos productivos actuales, en donde la automatización juega un rol muy importante

**OBJETIVOS ESPECÍFICOS**

* Introducir los sistemas automáticos flexibles como fruto de la necesidad de la sociedad industrializada.
* Desarrollar el pensamiento y la inteligencia mediante la práctica y el pensamiento creativo en la robótica
* Promover los experimentos, donde el equivocarse es parte del aprendizaje y el auto descubrimiento.
* Diferenciar las características de los sistemas mecánicos, automáticos y robóticos.
* Familiarizarse con el PC como sistema de control.
* Introducir los fundamentos de la programación de Robo Pro en los sistemas de control.

**ESTRATEGIAS METODOLÓGICAS**

***La metodología a utilizarse serán las siguientes:***

Participativa y creativa realizando actividades donde los estudiantes puedan adquirir conocimientos en la robótica y aportar sus conocimientos creativos y punto de vista diferente para diseñar y construir objetos robóticos con materiales tecnológicos como lo es el kit de Fischertechnik en conjunto con Robo Pro el cual permitirá programar logrando el funcionamiento de cada una de las piezas que componen el kit.

Se trabajara a través de modulo donde cada estudiantes podrá leer las clases siguientes y hacer apuntes de lo explicado en el seminario

El profesor explicara cada clase y resolverá las dudas de los estudiantes.

Los estudiantes programaran el robot que le servirán de ayuda para que puedan construir sus propios objetos mecanizados, logrando que desarrollen diferentes tareas.

Por cada actividad desarrollada los estudiantes deberán explicar el funcionamiento programado en Robo Pro de cada una de las funciones de estos objetos mecanizados.

**RECURSOS**

* + - Presentación en Power Point
		- Videos de Robots Fischertechnik
		- Kit de Fischertechnik
		- Laboratorio de computadoras multimedia
		- Internet
		- Correo Electrónico
		- RoboTxdroit ( celulares androind)
		- Robo Pro (software)

La asesoría fuera del horario de clases es otro aspecto fundamental para el desarrollo del curso. Los estudiantes llevan a esta actividad dudas concretas sobre aspectos tratados en clase o que en el estudio independiente no han sido comprendidos completamente.

Contenido

[La interfaz de control intelligent interface 8](#_Toc430115044)

[¿Qué es una interfaz de control? 13](#_Toc430115045)

[Las interfaces proporcionan 16](#_Toc430115046)

[Estructuras de programación de robots 18](#_Toc430115047)

[1. Posición de origen 18](#_Toc430115048)

[2. Inicialización de la variable contador de impulsos 19](#_Toc430115049)

[3. Contadores 21](#_Toc430115050)

[4. Programación orientada al estado 22](#_Toc430115051)

[ Comprensión de los elementos fischertechnik 29](#_Toc430115053)

|  |  |
| --- | --- |
| La interfaz de control intelligent interface |  |

##

Nos referiremos ahora a la interfaz de control intelligent interface de fischertechnik, cuyo esquema se muestra en la figura siguiente:

|  |
| --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00042/tema_2.3/fischer27.gif |

Sus características principales son, para el modelo 30402:

|  |  |
| --- | --- |
| Alimentación | Requiere:   \* fuente de alimentación con salida de 9V cc. y 1 A, Energy Set 30182 de fischertechnik -o cualquier alimentador universal de dichas características con el polo + en el interior del casquillo- (conexión en 1 -véase la figura superior-);   \* o 1 pila de 9 V;   \* o batería recargable "Accu Set" 34969 de fischertechnik  para uso autónomo con  robots móviles (Mobile Robot II) (conexión en los bornes denominados 2 en la figura)Posee limitador de corriente a 1A, resistente a cortos. |
| Puerto de comunicaciones | 1 puerto serie RS-232, con conector DB 9 (véase 4 en la figura) |
| Salidas | 4 digitales/de motor de 9V y 250 mA denominadas  M1, M2, M3, M4 (ver 5 en la fig.)Permiten inversión de giro pero no regulación de velocidad -esta interfaz se diferencia de la FlowGo en que no discrimina entre salidas digitales y salidas de motor- |
| Entradas | 8 digitales (sensores on/off), denominadas E1 - E8  (véase 6 en la fig.)   Sus características eléctricas son:   \* V = 6 - 12 V   \* Rentrada= 9 KW, aprox.   \* nivel de conmutación = 2.8 V2 analógicas (sensores de nivel), denominadas EX, EY (véase 7 en la fig.)   Sus características eléctricas son:   \* R = 0 - 5 KW   \* conversión analógico-digital de 10 bit y, por tanto, con 210=1024 niveles digitales y una resolución de 1023   \* tolerancia = 0.2% |
| Indicadores | 1 LED rojo indicador de encendido de la interfaz |
| Botones | Ninguno |
| Memoria | RAM de 32 kByte (para almacenamiento de un programa; si se interrumpe la alimentación se pierden los datos del mismo).EPROM de 64 kByte (sólo para el software del microprocesador) |
| Ampliación | Módulo de extensión para duplicar E/S digitales conectado mediante regleta de 14 polos (véase 8 en la fig.) |

Las clavijas de las E/S son de 2 mm, proporcionadas en los diferente modelos de montaje de la línea fischertechnik computing, en particular, en los conjuntos de control programado computing starter y en los conjuntos de robótica industry, pneumatic y mobile robots.

En la siguiente figura se puede observar un diagrama de bloques de la composición interna de la interfaz:



Puede funcionar en dos modos:

* modo en-línea (on-line, gobernada en todo momento conectada al PC por medio del puerto de comunicaciones), o
* modo descarga: en este modo el procesamiento del programa tiene lugar únicamente en el microprocesador de la interfaz, estando ésta desconectada del PC. En una primera fase se escribe el programa de control en el PC, luego se descarga en la interfaz y, finalmente, ésta puede cortar el contacto con el PC (de modo que el cable serie puede ser desconectado). La memoria de esta interfaz es de tipo RAM, de modo que si la interfaz es apagada, el programa descargado desaparece de la memoria.

La forma natural de programar la interfaz es el lenguaje LLWin de fischertechnik, pero en modo "on-line" puede ser utilizada mediante cualquier lenguaje de programación. Los detalles acerca de este proceso se pueden encontrar en [www.fischertechnik.de](http://www.fischertechnik.de). En dicha página figuran algunos drivers diseñados para diferentes lenguajes de programación. Flowol soporta la programación de esta interfaz sin necesidad de instalar drivers adicionales. Sin embargo, si se utiliza Flowol con esta interfaz, no se dispone del modo descarga.

Un ejemplo del conexionado de los actuadores y sensores propios de fischertechnik a la interfaz se muestra en la siguiente figura. En este ejemplo cuatro motores de un robot están conectados a las salidas digitales y ocho micropulsadores a las entradas digitales. De estos ocho, cuatro se utilizan como finales de carrera (E1, E3, E5, E7) y los otros cuatro (E2, E4, E6, E8) se utilizan como contadores de impulsos (golpeados por una rueda dentada de 4 dientes conforme giran los motores) que sirven para llevar la cuenta de la posición actual del robot - el conjunto motor de c.c. más contador de impulsos actúa como un motor paso a paso-. Este montaje se utiliza en un brazo de robot manipulador articulado de tres grados de libertad finalizado en una pinza.



Evidentemente la intelligent interface está diseñada específicamente para controlar los distintos dispositivos de control y robótica del fabricante fischertechnik. En particular, con los conjuntos Computing Starting y con los conjuntos de robótica industry, pneumatic y mobile robots. No obstante, un motor de c.c de entre 1.5 y 12 V como los utilizados en el aula-taller puede ser operado mediante la interfaz, aunque sus exigencias de corriente suelen ser mayores que las proporcionadas por la misma. De hecho, si se conecta este tipo de motor con cierta carga a la interfaz, se observa que el LED indicador de operación parpadea ligeramente en lugar de permanecer estable, lo que resulta indicativo de que el motor está exigiendo a la interfaz una corriente mayor que aquélla nominal para la que fue diseñada. Por otra parte, el tipo de conectores de 2 mm proporcionados por la interfaz no son excesivamente estándar en relación con los sensores y actuadores utilizados habitualmente en el aula-taller. No obstante esta interfaz resulta sumamente sencilla y "agradecida" de utilizar con los propios conjuntos de montaje de fischertechnik.

Como conclusión final podemos indicar que resulta recomendable utilizar la interfaz FlowGo con los proyectos de diseño y elaboración propia de los alumnos en el aula-taller y la intelligent interface con los conjuntos de montaje rápido y de robótica propios de fischertechnik, que agilizan notablemente el estudio de los diferentes aspectos del control y la robótica en el aula.

|  |  |
| --- | --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00042/tema_2.1/cont_enconor.jpg¿Qué es una interfaz de control? |  |

##

Un sistema de control debe estar equipado para la alteración de una o varias magnitudes de un sistema real con objeto de proporcionar una respuesta activa que modifique su comportamiento. Como ejemplo complejo de un sistema de control se puede pensar en un "edificio inteligente", que comprueba y analiza permanentemente el estado de la temperatura, la iluminación, etc., gracias a diferentes sensores. Mediante un ordenador es capaz de actuar de acuerdo con su programación sobre diversos dispositivos actuadores para regular las condiciones de las estancias y mantenerlas en los niveles deseados. Así pues, un sistema de control comprende, en general, los sistemas y dispositivos adelantados en el [tema 1.2](http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/Tema_1.2.htm) y que se esquematizan en la siguiente figura:



El primer elemento es el sensor o captador. Es un sistema que transfiere energía del sistema controlado (denominado normalmente planta), transformando la magnitud física que se mide (temperatura, presión, posición) en energía eléctrica (voltaje, corriente). Las resistencias dependientes de la temperatura, los fototransistores, los detectores de presión son sensores habituales. El sensor suele estar conectado a un circuito que genera una señal eléctrica de salida. Esta señal tiene frecuentemente características inadecuadas para su uso directo (baja intensidad, excesivo ruido, etc.), de modo que se debe recurrir a la utilización de un acondicionador de señal, cuya misión es adecuarla a las características de las etapas sucesivas y/o mejorar su calidad. Puesto que la mayoría de los sensores proporcionan señales analógicas, éstas necesitan ser convertidas en digitales para su posterior procesado. Esta función la desempeña un convertidor analógico-digital (A/D). Los datos digitales se utilizan en el procesador digital para que sean procesados de acuerdo con su programación. La señal digital ya procesada debe ser aplicada en el sistema real (planta), de modo que se necesita su reconversión en analógica, función que cumple el convertidor digital-analógico (D/A). A su salida, otro acondicionador la adecua a las necesidades específicas del elemento actuador que, utilizando energía eléctrica, neumática o hidráulica realiza alguna acción modificadora de las propiedades del sistema. Motores, electroválvulas, y lámparas son actuadores utilizados habitualmente.

Así pues, si queremos aprovechar un PC como cerebro de un sistema de control necesitamos acondicionar y convertir las señales que provienen de los sensores a las características de los puertos de entrada del PC, así como convertir y acondicionar las señales de los puertos de salida de éste a las características de los dispositivos actuadores.

En entornos educativos y de ocio se utilizan unos dispositivos denominados interfaces de control, o más prosaicamente controladoras, cuya misión es precisamente reunir en un solo elemento todos los sistemas de conversión y acondicionamiento que necesita un PC para actuar como cerebro de un sistema de control:



Las interfaces de control se podrían así definir como placas multifunción de E/S (entrada/salida) en configuración externa (es decir, no son placas instalables en ninguna bahía de expansión del PC), que se conectan con el PC mediante alguno de los puertos de comunicaciones propios del mismo (paralelo, serie o USB, generalmente) y sirven de interfaz entre el mismo y los sensores y actuadores de un sistema de control.

Las interfaces proporcionan

Las siguientes funciones:

1. entradas analógicas, que convierten niveles analógicos de voltaje o de corriente en información digital procesable por el ordenador. A este tipo de entradas se pueden conectar distintos sensores analógicos, como por ejemplo una LDR.
2. salidas analógicas, que convierten la información digital en corriente o voltaje analógicos de forma que el ordenador pueda controlar sucesos del "mundo real". Su principal misión es la de excitar distintos actuadores del equipamiento de control: válvulas, motores, servomecanismos, etc.
3. entradas y salidas digitales, usadas en aplicaciones donde el sistema de control sólo necesita discriminar el estado de una magnitud digital (por ejemplo, un sensor de contacto) y decidir la actuación o no de un elemento en un determinado proceso, por ejemplo, la activación/desactivación de una electroválvula.
4. recuento y temporización, algunas tarjetas incluyen este tipo de circuitos  que resultan útiles en el recuento de sucesos, la medida de frecuencia y amplitud de pulsos, la generación de señales y pulsos de onda cuadrada, y para la captación de señales en el momento preciso.

Algunas de las interfaces de control más avanzadas cuentan, además de con la electrónica precisa para el acondicionamiento y la conversión de las señales, con sus propios microprocesador y memoria. Así, con capaces hasta de almacenar pequeños programas de control transmitidos desde un PC que luego pueden ejecutar independientemente de su conexión a éste.

Algunas de ellas disponen también de bibliotecas de programación de las E/S para permitir su utilización con distintos lenguajes de propósito general, entre ellos, LOGO, BASIC y C.

Existen varios modelos comerciales, entre los que se pueden mencionar:

* Interfaz FlowGo, de [Data Harvest](http://www.data-harvest.co.uk)
* Interfaz intelligent interface de [fischertechnik](http://www.fischertechnik.com)
* Ladrillo RCX, de [Lego](http://www.lego.com)
* Interfaz Enconor, de [Enconor Tecnología Educativa](http://www.enconor.com)

|  |  |  |  |
| --- | --- | --- | --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00042/tema_2.1/cont_flowgo.gif | http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00042/tema_2.1/cont_fisch.gif | http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00042/tema_2.1/cont_rcx.jpg | http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00042/tema_2.1/cont_enconor.jpg |

Interfaces de control, de izquierda a derecha: FlowGo, intelligent interface, RCX, Enconor.

Puesto que en la dotación de nuestros centros educativos contamos con las dos primeras, dedicaremos los dos próximos temas de la presente unidad didáctica a estudiar las mismas con detalle.

|  |  |
| --- | --- |
| Estructuras de programación de robots | http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00046/tema_6.7/rob4fot.jpg |

# 1. Posición de origen

Para que un robot se dirija a su posición de origen no es necesario posicionarlo mediante el bloque posición, únicamente hay que dirigirlo en dirección al final de carrera que marca dicha posición y dejarle avanzar hasta que lo alcance.

La estructura es la siguiente:

|  |
| --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00046/tema_6.7/estruc1.gif |

#   2. Inicialización de la variable contador de impulsos

Cuando se lleva la articulación de un robot a una posición determinada, dicha posición se guarda en la variable contador correspondiente (por ejemplo en Z2). Si a continuación esa articulación se lleva a su origen hay que inicializar (poner a cero) dicha variable, ya que en caso contrario el sistema "piensa" que el robot sigue en su posición anterior.

|  |
| --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00046/tema_6.7/estruc2.gif |

Esta circunstancia se produce con frecuencia cuando se realiza un ciclo ya que, aunque el robot parte del reposo, en algún momento del programa se posiciona y retorna de nuevo al reposo.

|  |
| --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00046/tema_6.7/estruc4.gif |

En el caso excepcional de que la secuencia del programa sea lineal y sólo se pase por el origen al inicio del programa, no es necesario inicializar las variables contador, ya que al iniciar el programa todas las variables se ponen a cero.

|  |
| --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00046/tema_6.7/estruc5.gif |

Para evitar complicaciones en la mayoría de los casos la inicialización de la variable contador se introduce dentro del subprograma origen.

|  |
| --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00046/tema_6.7/estruc6.gif |

# 3. Contadores

En ciertas ocasiones es necesario ejecutar un proceso un número definido de veces. Una solución sería repetir el proceso ese número determinado de veces, pero además de agrandar innecesariamente el programa, en ciertas ocasiones no sería viable. Otra solución más inteligente es utilizar una variable contador en un bucle de tipo "HASTA" como la que se expone:

|  |
| --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00046/tema_6.7/estruc3.gif |

El proceso representado por un bloque se quiere repetir 10 veces. Primero se inicializa la variable contador, se ejecuta el proceso A una vez y la variable contador toma el valor 1, se comprueba si es igual a 10 y como no es así se repite de nuevo el ciclo hasta que se haya ejecutado 10 veces y la variable contador VAR1 sea 10.

# 4. Programación orientada al estado

En LLWin, el uso repetido de un mismo bloque funcional de subprograma incrementa la memoria requerida, porque cada vez que se inserta el bloque funcional del subprograma también se insertan internamente todos los bloques que se utilizan en dicho subprograma. Esto no suele suponer habitualmente un problema, pero si se crean proyectos muy grandes en un momento dado se pueden alcanzar los límites de LLWin o sobrecargar y ralentizar el microprocesador de la interfaz por exceso de bloques funcionales, en particular si se desea operar en el modo download. Por esta razón hay que intentar estructurar los programas de modo que los subprogramas sean invocados sólo una vez.

Para evitar en lo posible la llamada repetida a un mismo subprograma, se utiliza la denominada programación orientada al estado. Cada subprograma será insertado una sola vez y será llamado cuando corresponda en función del valor de una variable que determina el estado del sistema en cada instante. Vamos a ilustrar lo mencionado con un programa como el siguiente:

|  |
| --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00046/tema_6.7/estruc11.gif |

Este programa ejecuta de forma repetida varios procesos: A, B y C, aunque no siempre cada proceso se ejecuta en las mismas condiciones (observa que la variable Var2, que podría ser usada dentro de los procesos, cambia de valor). Podemos definir entonces varios estados relacionados con el orden de ejecución de los distintos procesos:

* Estado 0: Proceso A
* Estado 1: Proceso B
* Estado 2: Proceso A
* Estado 3: Proceso C
* Estado 4: Proceso B
* Estado 5: Proceso B
* Estado 6: Proceso C

Y utilizar una variable, por ejemplo Var1, para redistribuir el flujo del programa de forma que cada proceso se introduzca en el programa una sola vez.  El programa tendrá así la siguiente estructura:

|  |
| --- |
| http://platea.pntic.mec.es/vgonzale/cyr_0708/archivos/_15/html/mobligatorio/00046/tema_6.7/estruc12.gifComo se ha mencionado más arriba, esta forma de programar resulta especialmente adecuada en programas muy complejos en los que además se repiten varias veces llamadas a un mismo bloque de subprograma. |

**PRIMERA SEMANA**

**Competencias básicas:** Disposición para trabajar en equipo, indagación, respeto a las ideas, lectura, análisis, liderazgo y motivación por el diseño.

**Competencias Técnicas:** Manejo de básico de herramientas informáticas y construcción de creaciones propias.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OBJETIVOS | CONTENIDOS | TÉCNICA | ACTIVIDADES | RECURSOS | LOGROS DE APRENDIZAJE |
| * Introducir la interfaz de control intelligent interface para dar movimientos específicos a los robots.
 | La interfaz de control intelligent interfaceCaracterísticas principales | Exposición dialogada | Conversar sobre los contenidos y evaluación del curso.Demostración de los control intelligent interface | TableroMarcadorR. HumanoData showInternet  | Conocer los puertos de control intelligent interface y sus funciones. |
| Comprender el funcionamiento del modo en línea y el modo descarga. | Conceptos y definiciones de la interfaz control intelligent interface | Conversatorio | Observar la función de modo en línea y el modo descarga. Investigar sobre los avances de la robótica en nivel universitario | ComputadoraKit fischertechnikControl intelligent interface | Conocer el funcionamiento del modo en línea y el modo descarga. |
| Conocer el uso de motores y bombillos | MotorMotores xsBombillos | Exposición de las piezas y su uso con el controlador | Observar el uso de los motores en los diferentes puertos | -Kit de fischertechnik.-Control intelligent interface-computadora | Aprender el uso del controlador |
| Conocer la interfaz gráfica de Robo Pro. | InicioFinBifurcación digitalBifurcación analógicaTiempo de esperaSalida del motorMotor decodificadorEspera a entradaContador de impulsoBucle contador | Explicación de los diferentes elementos del programa y su función. | Identificar los elementos de Robo Pro | -Kit de fischertechnik.-Control intelligent interface-computadora-Robo Pro | Instruirse con los elementos de Robo Pro. |

**Trabajo a Distancia:**

* + - Sobre el uso de intelligent interface en los dispositivos de Control Computing, Pneumatic y movile Robots.

**SEGUNDA SEMANA**

**Competencias básicas:** Disposición para trabajar en equipo, indagación, liderazgo, motivación por el diseño, lectura y análisis.

**Competencias Técnicas:** Manejo de básico de herramientas informáticas y construcción de creaciones propias.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OBJETIVOS | CONTENIDOS | TÉCNICA | ACTIVIDADES | RECURSOS | LOGROS DE APRENDIZAJE |
| * Realizar un abanico, que este programado para dar vueltas en una velocidad 5 y gire por 30 segundos
 | * Programación en Robo Pro
 | Trabajo en equipo | Programación utilizando: inicio, fin, motor decodificador, tiempo de espera. | -Kit de fischertechnik-Control intelligent interface-computadora-Robo Pro | Plasmar en el alumno conocimientos significativos sobre la programación en la robótica, al igual que ver plasmado la programación en su prototipo.  |
| * Crear la programación de un semáforo, con duración de 20 segundos por cada bombillo, usar un pulsador para enviar la señal de inicio
 | * Programación en Robo Pro
 | Trabajo en equipo | Programación utilizando: inicio, fin, salida de motor, tiempo de espera, bucle contador, esperar a entrada. | -Kit de fischertechnik-Control intelligent interface-computadora-Robo Pro | Identificar como se programan diferentes aparatos tecnológicos con Robo Pro. |

**Trabajo a distancia:**

* + - Con el software instalado en sus computadoras programar el uso de una puerta electrónica, que abra cada 5 segundos y permanezca abierta por 15 segundos, repitiendo esto 10 veces

**TERCERA SEMANA**

**Competencias básicas:** Disposición para trabajar en equipo, indagación, liderazgo, motivación por el diseño, lectura y análisis.

**Competencias Técnicas:** Manejo de básico de herramientas informáticas y construcción de creaciones propias.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OBJETIVOS | CONTENIDOS | TÉCNICA | ACTIVIDADES | RECURSOS | LOGROS DE APRENDIZAJE |
| * Realizar un montacargas con las piezas del Kit de fischertechnik.
 | Comprensión de los elementos fischertechnik* Bloques de construcción
* Lámpara esférica
* Lámpara de lente
* Fototransisto
* Motor
* Pulsador
* Sensor
* Engranaje
* ROBOTICS TXT Controller
 | Participación colectivaTrabajo en equipo | - Crear un montacargas, con bombillos, actuadores y pulsador. | -Kit de fischertechnik-Control intelligent interface-computadora-Robo Pro | Adquirir conocimientos sobre la creación de aparatos automatizados. |

**Trabajo a distancia:**

* + - Con el software instalado en sus computadoras programar el uso de una puerta electrónica, que abra cada 5 segundos y permanezca abierta por 15 segundos, repitiendo esto 10 veces.

**CUARTA SEMANA**

**Competencias básicas:** Disposición para trabajar en equipo, indagación, liderazgo, motivación por el diseño, lectura y análisis.

**Competencias Técnicas:** Manejo de básico de herramientas informáticas y construcción de creaciones propias.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| OBJETIVOS | CONTENIDOS | TÉCNICA | ACTIVIDADES | RECURSOS | LOGROS DE APRENDIZAJE |
| Realizar una conferencia con los grupos de informática de Isae Universidad. | * Todo los contenidos dados en el seminario de Robótica de ensamblaje y programación.
 | Exposición  | - Explicación de la robótica en la actualidad.- Demostración y explicación de cada pieza y su función- Montaje de las piezas-Explicación de los prototipos realizados -Explicación de la programación-demostración de los prototipos con la programación. | kit de fischertechnik.VideosManual de ensamblaje. | Exponer todos los conocimientos adquiridos con los estudiantes de la Lic. en Informática. |

**Bibliografía**

* Tecnologías de la Información y de la Comunicación. Capítulo 6, Programación y control de procesos. Juan A. Alonso, Santiago Blanco A., Santiago Blanco S., Roberto Escribano, Víctor R. González, Santiago Pascual, Amor Rodríguez. Editorial Ra-Ma, 2004.

* [Control y Robótica. Tema: La interfaz de control](http://cfievalladolid2.net/tecno/ctrl_rob/flowol/interfaz.htm). Curso provincial. Víctor R. González, Asesoría de Tecnología y FP, CFIE Valladolid II.
* Michael Brady and Richard Paul, editors. Robotics Research: The First International Symposium. The MIT Press, Cambridge MA, 1984
* [Control y Robótica. Tema: Fundamentos de robótica](http://cfievalladolid2.net/tecno/ctrl_rob/robotica/index.htm). Curso provincial. CFIE Valladolid II. Víctor R. González. Asesoría de Tecnología y FP.
* [Robótica](http://proton.ucting.udg.mx/materias/robotica/). Universidad de Guadalajara. Méjico.
* Tecnologías de la Información y de la Comunicación. Capítulo 6, Programación y control de procesos. Juan A. Alonso, Santiago Blanco A., Santiago Blanco S., Roberto escribano, Víctor R. González, Santiago Pascual, Amor Rodríguez. Editorial Ra-Ma 2004.