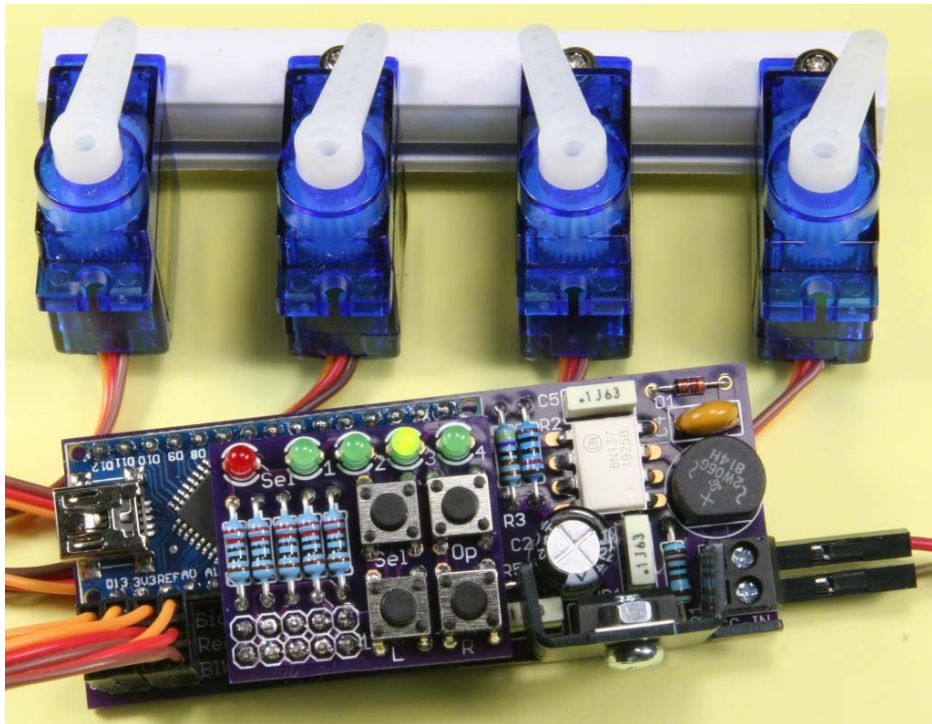


Construya su propio decodificador Quad Servo DCC - Parte 2

Introducción

Tal y como se explica en la Parte 1 de este artículo, el Decodificador Quad Servo DCC (QSDD), basado en un módulo Arduino Nano estándar, es un decodificador de accesorios que puede construir usted mismo por menos de \$ 20, y que es capaz de controlar con precisión la posición de hasta cuatro servos. Si siguió los detalles de construcción dados en la Parte 1, y ha construido su propio QSDD y cargado su software ("sketch"), entonces esta parte final cubre todos los detalles de cómo configurar los movimientos y velocidad de cada servo con precisión, y cómo asignar direcciones a los servos para que puedan ser controlado a través del sistema DCC que posea. Finalmente, hay algunos detalles técnicos de la QSDD, más complejos, para aquellos lectores que estén interesados.

Para el resto del artículo, se asume que los dos módulos QSDD, el decodificador y el teclado, están conectados, ya sea directamente (como se muestra a continuación) o mediante un cable plano, y que los cuatro los servos están conectados a los pines apropiados.



Todas las fotografías del autor

El QSDD debe alimentarse desde su sistema DCC mediante una conexión a la vía o directamente a la central desde la que también puede recibir comandos de accesorios DCC estándar.

Los comandos se pueden enviar utilizando controladores portátiles o software adecuado, como JMRI Panel Pro (<http://jmri.org/help/en/html/apps/PanelPro/index.shtml>) o mi propia aplicación A-Track (<https://www.a-train-systems.co.uk/atrack.htm>), ejecutándose en su ordenador.

El teclado también se puede utilizar para accionar los servos a mano, utilizando los botones instalados, ya sea cuando se conecta directamente al decodificador o, mejor, a través de un cable plano de hasta 40 pulgadas largo. Este último le permite ver y configurar sus desvíos cuando el decodificador y los servos están fuera de la vista bajo la maqueta.

Tenga en cuenta que, una vez configurado, si elige operar los servos solo a través de comandos DCC, entonces no es necesario tener el teclado conectado al decodificador.

Configuración del servo paso a paso:

Aunque una conexión USB no tiene ningún efecto sobre el proceso de configuración en sí, puede llevarse a cabo ya sea que haya uno presente o no, si tiene el cable USB del ordenador conectado al Arduino Nano con el IDE de Arduino en ejecución (tenga el boceto QSDD cargado o no), entonces los mensajes para confirmar todos los cambios realizados durante la configuración se pueden mostrar en la pantalla de su ordenador a medida que ocurren. Esto puede resultar muy útil, especialmente al realizar la configuración por primera vez.

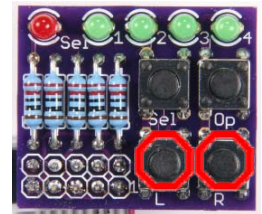
La visualización de mensajes solo requiere que el Monitor Serie esté habilitado en el IDE de Arduino haciendo clic en el icono correspondiente (🔍) en el extremo derecho de la barra de herramientas, con la velocidad establecida en 115200 baudios, dejando los mensajes DEBUG habilitados en el sketch (que es como se proporciona el código).

Nota.- Una vez que se ha cargado el sketch QSDD, siempre que se conecte el cable USB al Arduino Nano verá que el LED rojo **Sel** y uno o más de los LED verdes parpadearán unas pocas veces (¹) ya que la conexión USB está completamente establecida, dependiendo de la versión del IDE de Arduino instalado en su ordenador. Una vez que el enlace USB sea estable, la placa decodificadora se reiniciará automáticamente, encendiendo, y luego apagando, todos los LED una vez más, y moviendo los servos a su última posición definida.

Paso 1

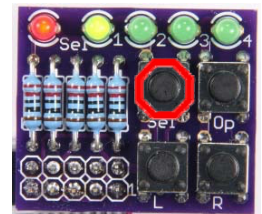
El primer paso es encontrar el punto central del movimiento de cada servo. Micro servos estándar como el SG90 tienen un rango nominal de 180 grados, desde el máximo a la derecha a 0 ° hasta el máximo a la izquierda a 180 ° aunque, debido a que están contruidos con tolerancias bastante amplias, el rango puede variar hasta en 15°.

Suponiendo que tiene el decodificador y el teclado conectados juntos, alimentado el conjunto, y el software cargado, presione juntos los botones **L** y **R** del teclado, luego suéltelos. Todos los servos deberían moverse a sus posiciones centrales (90°). A continuación, puede ajustar el montaje de los brazos servoactuadores, si es necesario, para que se alineen con cualquier posición que desee como centro de rango de cada servo.



Paso 2

Presione y suelte el botón Seleccionar (**Sel**): el LED rojo **Sel** se iluminará junto con el LED verde **1**, y el servo 1 se moverá a su límite izquierdo desde su actual posición. Ahora puede ajustar la posición límite izquierda hacia la izquierda, un grado cada vez, presionando el botón izquierdo (**L**), o hacia la derecha presionando el botón derecho (**R**). Mantenga presionados los botones **L** o **R** hasta que alcanzar la posición límite izquierda requerida por la conexión mecánica al desvío que tiene la intención de utilizar.



Paso 3

Presione y suelte el botón **Sel** nuevamente: los LED rojo **Sel** y verde **1** permanecen encendidos, pero el servo 1 ahora se moverá a su posición límite derecha actual que se puede ajustar, como se indica arriba en el Paso 2, presionando los botones **L** o **R**.

Paso 4

Presione y suelte el botón **Sel**: el rojo **Sel** y el LED verde **1** permanecen encendidos y el servo 1 ahora se moverá, a su velocidad establecida, a su posición límite izquierda, luego regresará a la misma velocidad a su posición límite derecha. Si la velocidad de tránsito es demasiado rápida o demasiado lenta para sus requisitos, presione el botón **L** para reducir la velocidad (despacio), o el botón **R** para acelerar la velocidad (rápido). Después de cada pulsación del botón, el servo se moverá de derecha a izquierda y volverá a la nueva tasa. Tenga en cuenta que el decodificador no responderá a las pulsaciones de botones mientras un servo está en movimiento, pero que puede mantener presionado cualquiera de los botones continuamente y observar el tránsito de velocidad para el servo 1 ya sea para acelerar o desacelerar hasta que alcance la velocidad deseada.

1.- Importante: el comportamiento de los LED se refiere a la versión 4.6. En la **versión 5,2** la secuencia de encendido es diferente: ver **"Construya su propio decodificador Quad Servo DCC. Actualización"**

Tenga en cuenta que, cuando se haya alcanzado la velocidad más lenta o más rápida, el movimiento del servo se detendrá y el decodificador no responderá a más pulsaciones de cualquier botón con el que alcanzó su punto más bajo o el límite de tasa superior.

Paso 5

Presione y suelte el botón **Sel**; esta vez el LED rojo **Sel** parpadeará a un ritmo constante, con LED verde **1** encendido. Ahora tiene la opción de guardar todos los ajustes de posición y ajustes que ha creado como valores de variable de configuración (CV), reemplazando la configuración predeterminada original. Estos valores CV guardados se mantendrán permanentemente en la memoria Arduino Nano incluso cuando esté apagado.

Para guardar los nuevos valores de CV para el servo 1, presione el botón **R** (Retener los cambios y Reemplazar los valores CV actuales). Alternativamente, si no ha realizado ningún cambio en los parámetros del servo 1, o desea descartar los ajustes que realizó, presione el botón **L** en su lugar (Perder cambios y Dejar los valores CV actuales como están). Una vez guardados o descartados los cambios, el LED verde 1 se apagará, aunque el LED rojo **Sel** seguirá parpadeando.

Nota.- También tiene la opción, en el Paso 5, de presionar y soltar el botón **Sel** en lugar de los botones **L** o **R**. Esto lo llevará inmediatamente al Paso 6, como se describe a continuación, y descarta cualquier cambio que haya realizado en los parámetros del servo 1 (como presionar la tecla **L**).

Paso 6

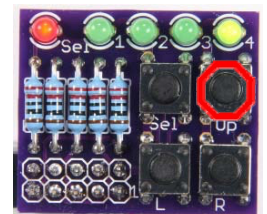
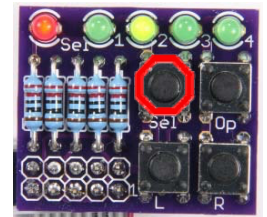
Presione y suelte el botón **Sel** una vez más para completar el proceso la configuración para el servo 1 (a menos que salte aquí directamente desde el Paso 5). El LED rojo **Sel** dejará de parpadear, mientras que el LED verde **1** se apagará y el verde **2** se iluminará en su lugar. Ahora puede repetir las acciones de los pasos 2 al 6 para llevar a cabo la configuración de los servos 2, 3 y 4. Después de haber completado la configuración del servo 4, presionando y soltando el botón **Sel** terminará el proceso de configuración y todos los LED se apagará.

Si desea realizar la configuración de servos individualmente más tarde, en lugar de los cuatro, en el Paso 2 o en el Paso 6, presione y suelte el botón **Op**. Esto omitirá la configuración del servo seleccionado actualmente, apagando el LED verde correspondiente e iluminando el LED verde del siguiente servo. Presionar el botón **Op** cuando se ha seleccionado el servo 4 terminará el proceso de configuración completo (que luego, por supuesto, puede reiniciarse en servo 1 presionando y soltando el pulsador **Sel**).

Una vez que tenga el rango de movimiento de cada servo ajustado a lo que considera adecuado, puede ajustar los servos en los montajes y enlaces mecánicos que esté utilizando para conectarse a los desvíos de la maqueta. Después de que todo esté asegurado, repita el proceso de configuración del servo para ajustar los límites de posición y velocidad de movimiento.

Nota.- Después de que un servo llega a la posición límite requerida, se “*desconecta*”, lo que significa que ya no está impulsado activamente por el decodificador y que sólo mantiene la fricción en el tren de engranajes del servo el enlace mecánico con el desvío. Por lo tanto, puede optar por mover las posiciones límite un grado o dos para asegurar que los puntos de desvío activados permanezcan en su posición. El problema no surge con Desvíos de Peco, por ejemplo, que tienen un resorte sobre el centro para mantener la posición conmutada.

Un [breve video](#) ⁽¹⁾ de la secuencia de configuración del decodificador está disponible en las páginas de Model Railroad Hobbyist en YouTube.

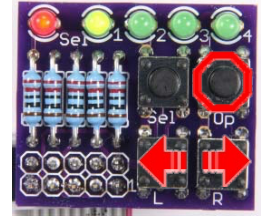


1. Aunque el vídeo está en inglés, visualmente es lo suficientemente claro en la exposición del funcionamiento del decodificador.

Operación del servo - Manual

El funcionamiento correcto de los servos conectados se puede comprobar pulsando **Op** en el teclado. Esto encenderá el LED verde **1** para mostrar que se ha seleccionado el servo 1.

Presione el botón **L** para mover el servo 1 a su posición límite izquierda, en el ajuste velocidad, o el botón **R** para mover el servo 1 a su posición límite derecha. No habrá movimiento, por supuesto, si el servo ya está posicionado en el límite seleccionado.



Al presionar más el botón **Op**, se encenderán los LED verdes **2, 3 y 4**, a su vez, seleccionando los correspondientes servos 2, 3 y 4, y separando el servo previamente seleccionado. Cuando el correspondiente LED verde está iluminado, el servo seleccionado actualmente se puede operar usando los pulsadores **L y R**, como se describe.

Una última pulsación del botón **Op** apagará todos los LED, con todos los servos desconectados.

La conexión del teclado a la placa decodificadora mediante un cable plano de 10 vías con los conectores permitirá montar el teclado en una *fascia* (marco exterior de la maqueta) para el control directo de sus desvíos, si así lo desea. Se puede usar un cable de hasta 40 pulgadas (1 metro, aprox.) de largo.

Operación del servo - Comandos DCC

Cuando el decodificador está conectado a la vía de su maqueta, puede operar cualquiera de los servos (y el desvío al que esté vinculado) emitiendo el comando apropiado desde su sistema DCC para cualquier dirección que se haya asignado a ese servo. Se asignan las direcciones predeterminadas 1, 2, 3 y 4 como parte del software precargado, pero posteriormente puede asignar las direcciones que desee a los cuatro servos, como se explica en la siguiente sección sobre la configuración de las direcciones de salida del decodificador.

Las secuencias de teclas para operar los servos, y sus desvíos adjuntos, se describen a continuación para algunos de los controladores de mano comunes del sistema DCC:

- | | |
|-----------------|---|
| NCE | Presione 'SELECT ACCY' Escriba una de las direcciones configuradas seguido de 'ENTER'. Presione '1' o '2' para mover el desvío (dependiendo de su posición actual). Si el desvío no se mueve, presione 'SELECT ACCY' dos veces para moverlo en el dirección opuesta. |
| Digitrax | Presione 'SWCH'. Escriba una de las direcciones configuradas. Presione 'OPTN / t' o 'CLOC / c' para mover el desvío (dependiendo de su actual posición). |
| Lenz | Presione 'F' y luego '5' (LH100) o 'Points / Signals' (LH101). Escriba una de las direcciones configuradas seguido de 'ENTER' o 'Points / Signals' 'Presione '+' o '-' (LH100) or 'M' (LH101) para mover el desvío (dependiendo de su posición actual). |
| MRC | Presione 'ACCY'. Escriba una de las direcciones configuradas seguido de 'ENTER' Presione '1' o '2' para mover el desvío (dependiendo de su posición actual). |

Cuando se utilizan comandos DCC para operar los servos, no es necesario tener el teclado instalado en la placa decodificadora, ni una conexión USB al módulo Arduino Nano. La placa decodificadora por sí sola proporciona todas las funciones de un decodificador de accesorios estándar.

Después de completar cada operación, ya sea iniciada manualmente o mediante un comando DCC, la posición del servo involucrado se almacena en la variable de configuración correspondiente (CV59, 64, 69 o 74).

Las posiciones se conservan cuando el decodificador se apaga de modo que, cuando se enciende de nuevo, los servos se repositionarán en el lugar donde se les ordenó por última vez.

Configuración de direcciones de servo (desvío)

Si bien las cuatro direcciones predeterminadas (1, 2, 3 y 4) pueden ser adecuadas si su diseño solo tiene cuatro desvíos, normalmente será necesario asignar diferentes direcciones individuales al decodificador para cada servo.

Para comenzar a asignar direcciones, coloque un puente (*jumper*) entre los dos pines del terminal de programación en la placa del decodificador (etiquetados "Address"). Para la siguiente etapa, la conexión del teclado es opcional, por lo que asumiremos que se ha eliminado para encajar el puente y se ha dejado desconectado.

Desde su sistema DCC prepárese para enviar un comando de accesorio al decodificador, como se describe en la sección anterior, seleccionando la función Accesorio para su tipo de sistema DCC.

Escriba la dirección que desea asignar al servo 1, seguida de la tecla ENTER si es necesario, después cancele el desvío. Usted puede seleccionar cualquier dirección de desplazamiento ya que la dirección del comando es ignorada, y sólo se registra la dirección utilizada.

Para ser aceptadas, las direcciones deben estar en el rango de 1 a 2043, es decir, se aceptan direcciones de dos o cuatro dígitos. Los valores de dirección fuera de este rango simplemente se ignorarán y no reemplazarán cualquier valor que esté almacenado actualmente para la dirección de servo (desvío) correspondiente. Ahora seleccione la función Accesorio nuevamente en su controlador de mano, introduzca la dirección que desea asignar al servo 2, y proceda a mover el desvío. Repita esta secuencia dos veces más para programar las direcciones requeridas para los servos 3 y 4.

Las cuatro direcciones introducidas ahora se almacenarán en las variables de configuración 41 a 48, con cada dirección retenida en dos ubicaciones consecutivas de CV, por ejemplo: dirección del servo 1 en CV41 y CV42, servo 2 dirección en CV43 y CV44, etc.

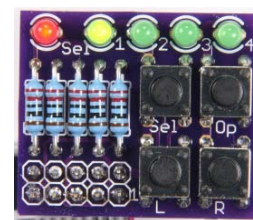
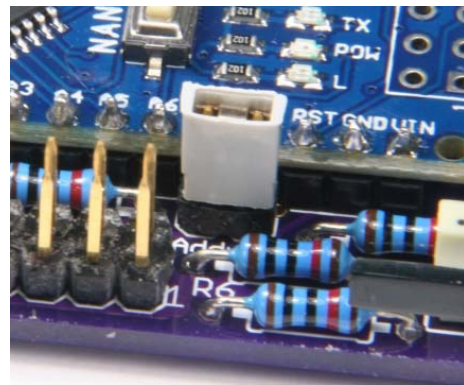
Retire el puente del cabezal de programación de dos pines para terminar la configuración de direcciones de servo. Esta acción reiniciará el decodificador y luego permitirá que funcione normalmente. El puente se puede almacenar en uno de los pines del cabezal de programación hasta la próxima vez que se requiera.

Nota.- Si tiene una conexión USB al Arduino Nano desde su PC y el Monitor Serial de Arduino IDE está habilitado (como se describió anteriormente en la sección Configuración del servo), los mensajes de confirmación (DEBUG) para cada paso de programación de la dirección del servo se mostrarán en la ventana Monitor Serial.

Monitoreo y control manual de la configuración de direcciones de servo

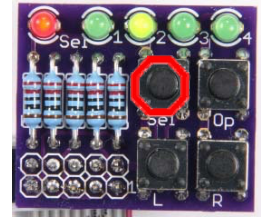
Conectando el teclado a la placa decodificadora (mejor a través de un cable) cuando el puente que cortocircuita Address se coloca en los dos pines, se ofrece más flexibilidad en la asignación de direcciones y también proporciona información visual a través de los LED del teclado durante el proceso de programación.

Con el puente en su lugar, el LED rojo **Sel** y el LED verde **1** se encenderán. Configurar la dirección para el servo 1 utilizando su controlador de mano como se describe anteriormente, hará que el LED rojo **Sel** parpadee indicando que la dirección ha sido recibida, el LED verde **1** se apagará y el LED verde **2** se encenderá. Posteriormente, ingresar direcciones para los servos 2, 3 y 4. Repita la secuencia con los LED verdes 2, 3 y 4. Una vez recibidas las cuatro direcciones, se almacenarán en CV 41 al 48, todos los LED verdes se apagarán y el LED rojo **Sel** parpadeará continuamente.



En este estado, el decodificador no responderá a ninguna pulsación de botón ni a ningún comando DCC. Quite el puente (Address) para terminar la programación, apagando el LED rojo **Sel**. Esto reiniciará decodificador y vuelve a ponerlo en funcionamiento normal.

Sin embargo, con el teclado conectado y el puente Addr en su lugar, tiene la opción de elegir qué direcciones de servo programar. Comience con LED rojo **Sel** y verde **1** encendidos, presionando el pulsador Select (**Sel**) apagará el LED verde **1** y se encenderá el LED verde **2**, y omitirá la entrada de una nueva dirección para el servo 1. Ahora puede introducir una nueva dirección para el servo 2 u omitir esto también presionando el botón **Sel** de nuevo, que pasará al LED verde **3** en lugar del LED verde **2** y le ofrece la oportunidad de programar una nueva dirección para el servo 3.



Cuando llegue a la opción de programar una dirección para el servo 4, ingrese una nueva dirección (o presione **Sel** para omitir la programación) apagará todos los LED verdes, almacenará cualquier dirección ingresada en la CV correspondiente y quedará el LED rojo **Sel** parpadeando continuamente. Como se dijo antes, en este estado el decodificador no responderá a ninguna pulsación de botón ni a ningún comando DCC. Quitar el puente para terminar la programación, apagar el LED rojo **Sel**, reiniciar el decodificador para devolverlo al funcionamiento normal. Guarde el puente en uno de los pines del grupo de conexión de programación hasta la próxima vez que lo necesite.

Nota.- Como se mencionó anteriormente, si tiene una conexión USB al Arduino Nano desde su PC, los mensajes de confirmación (DEBUG) para cada paso de programación de la dirección del servo se mostrarán en el ventana Monitor Serial.

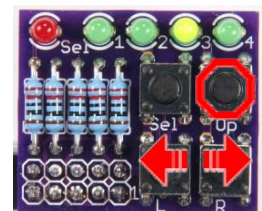
Inversión de la dirección del servo

Después de montar los servos en los desvíos de su maqueta, es posible que necesite que algunos operen en la dirección opuesta, de modo que la generación de un comando de ruta desde su sistema DCC cambie el desvío a la dirección de la Ruta (desviado) en lugar de configurarlo en la dirección Normal (recto).

Para cambiar la dirección de funcionamiento de un servo, conecte el teclado al decodificador y presione el botón **Op** una o más veces para encender el LED verde correspondiente al servo a invertir.

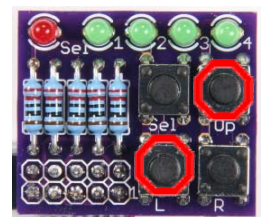
Ahora presione y suelte el botón **Sel una vez**. El LED rojo **Sel** se iluminará brevemente para reconocer que el cambio requerido se ha guardado en el variable de configuración involucrada (CV56, 61, 66 o 71), después de lo cual comprobará que presionando el botón **L** moverá el servo seleccionado a la derecha, y el botón **R** lo mueve hacia la izquierda. El servo reaccionará de manera similar si recibe comandos de su sistema DCC.

Complete el proceso presionando el botón **Op** tantas veces como sea necesario para apagar todos los LED verdes. Exactamente la misma secuencia de acciones devolverá el servo a la normalidad en lugar de revertir operación.

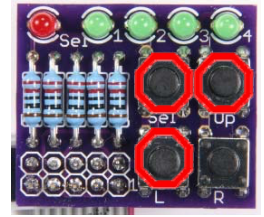


Reset del decodificador

Como se mencionó anteriormente, hacia el final del "Software para QSDD" en la Parte 1 de este artículo, para resetear el decodificador y reiniciar el software cargado, normalmente presionaría el botón de reinicio en el Arduino Nano. Sin embargo, con el teclado conectado directamente a la placa del decodificador, el botón de reinicio no es de fácil acceso, por lo que, como alternativa, simplemente presione y mantenga presionado el botón **L** en el teclado; seguidamente presione el pulsador **Op**, luego suelte ambos botones para reiniciar el Arduino Nano. Todos los LED deben encenderse y luego apagarse, y todos los servos conectados deben moverse a (o permanecer en) su último comando de posición.



También puede, si lo desea, restaurar todos los parámetros del decodificador a sus valores predeterminados (consulte la lista de variables de configuración (CV) proporcionada en la Parte 2 de este artículo). Con el teclado enchufado o conectado a la placa del decodificador, presione y mantenga presionado el botón **L** en el teclado. Ahora presione y mantenga presionado el botón **Op**; luego, con ambos botones presionados (y un poco de destreza manual), presione y suelte el botón **Se1**. Finalmente, suelte los botones **L** y **Op**.



Al igual que con el reinicio normal, todos los LED deben encenderse y luego apagarse.

Sin embargo, esta vez, el LED rojo **Se1** se iluminará brevemente de nuevo cuando se carguen los valores predeterminados de CV seguido de que todos los servos conectados se mueven en el sentido de las agujas del reloj a su posición predeterminada más a la derecha (20 grados a la derecha del centro).

Tenga cuidado, si los servos están conectados mecánicamente a un desvío de la maqueta, que este movimiento se pueda realizar sin causar daños en la articulación ni en el servo, teniendo en cuenta que el plástico del tren de engranajes instalado en la mayoría de los servos se daña fácilmente si el servo se mueve contra un tope final.

Variables de configuración y operaciones DCC

Como todos los decodificadores compatibles con NMRA, el QSDD mantiene sus datos de trabajo en un conjunto de variables de configuración (CV) que constan de:

| CV No. | Default Value | Description |
|--------|---------------|---|
| 01 | 1 | Board Address LSB – internal use – ignore any value loaded here |
| 07 | 89 | NmraDCC Version |
| 08 | 13 | Manufacturer (Do-It-Yourself) |
| 09 | 0 | Board Address MSB – internal use – ignore any value loaded here |
| 29 | 226 | Decoder Configuration – Extended Accessory + Output Addressing |
| 41 | 1 | Output 1 Address LSB |
| 42 | 0 | Output 1 Address MSB |
| 43 | 2 | Output 2 Address LSB |
| 44 | 0 | Output 2 Address MSB |
| 45 | 3 | Output 3 Address LSB |
| 46 | 0 | Output 3 Address MSB |
| 47 | 4 | Output 4 Address LSB |
| 48 | 0 | Output 4 Address MSB |
| 50 | 0 | Load Default CV Values if Not = 173 (0xAD), Auto set = 173 after load |
| 55 | 6 | Servo 1 - Slow Rate (1 to 16) |
| 56 | 1 | Direction - 1 = Normal Operation, 0 = Reverse Operation |
| 57 | 70 | Right Limit |
| 58 | 110 | Left Limit |
| 59 | 70 | Current Position |
| 60 | 6 | Servo 2 - Slow Rate (1 to 16) |
| 61 | 1 | Direction - 1 = Normal Operation, 0 = Reverse Operation |
| 62 | 70 | Right Limit |
| 63 | 110 | Left Limit |
| 64 | 70 | Current Position |
| 65 | 6 | Servo 3 - Slow Rate (1 to 16) |
| 66 | 1 | Direction - 1 = Normal Operation, 0 = Reverse Operation |
| 67 | 70 | Right Limit |
| 68 | 110 | Left Limit |
| 69 | 70 | Current Position |
| 70 | 6 | Servo 4 - Slow Rate (1 to 16) |
| 71 | 1 | Direction - 1 = Normal Operation, 0 = Reverse Operation |
| 72 | 70 | Right Limit |
| 73 | 110 | Left Limit |
| 74 | 70 | Current Position |

Además de este “conjunto de trabajo”, hay otras tres CV que controlan funciones “especiales”:

| CV No. | Default Value | Description |
|--------|---------------|---|
| 120 | 0 | Set = 120 to load Default CVs – must be cleared manually |
| 121 | 24 | Address to set CV Values in Operations Mode (Program on Main) – LSB |
| 122 | 0 | Address to set CV Values in Operations Mode (Program on Main) – MSB |

Veamos primero las funciones “especiales”, la dirección contenida en las CV 121 y 122 se usa para escribir un nuevo valor en cualquier CV dentro del decodificador usando el Modo de Operaciones (también llamado Programación en la Principal). Configure su sistema DCC a través de su controlador de mano en modo de operaciones y prepárese para usar una dirección de locomotora –no una Dirección de Accesorio– igual a la dirección contenida en las CV 121 y 122, que es 24 por defecto.

Seleccione el número de CV que se programará, ingrese el nuevo valor y presione ENTER (o la tecla destinada por su sistema para completar el comando). Dado que no es posible volver a leer valores de CV en el modo de operaciones, tendrá que juzgar, por el comportamiento posterior del decodificador, si el cambio de valor de CV tuvo éxito.

La otra función “especial”, iniciada programando un valor de 120 en CV120 (a través de Modo Operaciones), es manejada por la biblioteca NmraDcc y carga todas las CV con sus valores predeterminados cuando el decodificador se reinicia o se enciende. El restablecimiento se logra presionando el botón Reset en el Arduino Nano o, manteniendo presionados los botones **L** y **Op** en el teclado y luego liberándolos.

Sin embargo, parece que CV120 no es borrada automáticamente por la biblioteca NmraDcc, por lo que las CV continuarán restableciéndose a sus valores predeterminados cada vez que se reinicie el decodificador hasta que cambie el valor en CV120 usted mismo.

Para evitar este inconveniente, el QSDD se ha configurado para utilizar CV50 como alternativa. Algún valor *excepto* 173 en esta CV restablecerá todas las CV a sus valores predeterminados la próxima vez que se reinicie o encienda el decodificador. Después de lo cual el valor de CV50 se establecerá en 173 (hex AD = “all Default”) automáticamente. Esto significa que la carga predeterminada solo ocurre una vez y ocurrirá automáticamente cuando el sketch del decodificador se carga en el Arduino Nano por primera vez (cuando normalmente todos los CV contienen el valor 255). Si está conectado a una Vía de Programación, el QSDD, a diferencia de la mayoría de los decodificadores de accesorios comerciales, le permite leer los valores actuales de todos los CV y luego escribir (programar) nuevos valores si es preciso. Cuando haga esto, debe asegurarse de que el teclado esté conectado al decodificador, ya que el reconocimiento necesario de retorno a la estación de comando DCC (un pulso de corriente) es generado al encender brevemente los cinco LED del teclado juntos.

La lectura y escritura de los valores de CV a través de una Vía de programación (Modo de Servicio) se puede realizar mediante un controlador de mano conectado a todos los tipos principales de sistemas DCC, pero (a menos que alguien pueda decirme lo contrario, no de JMRI Decoder Pro, que simplemente decide que el decodificador no es una locomotora y se niega a seguir adelante).

Sin embargo, si tiene algún tipo de sistema NCE DCC más una computadora con Windows, mi propia aplicación A-Track (www.a-train-systems.co.uk/atrack), leerá y programará con gusto los CV de QSDD y le permitirá guardar un registro de ellos en un archivo.

Nota.- Si bien un sistema NCE Power Cab programará felizmente el QSDD en cualquier modo, el sistema más antiguo NCE Power Pro tiene un hardware de programación mucho menos tolerante. Si desea utilizar un Power Pro para leer o escribir CV de QSDD, debe conectarse al QSDD a través de un amplificador de programación como SoundTraxx PTB100, o hacer que el módulo Arduino Nano se alimente desde un puerto USB del ordenador mientras se realiza la programación. Si utiliza el modo de programación directa desde el Power Pro da como resultado demasiados errores (valores de CV mal leídos), por lo que se recomienda el modo Página. (**Con la versión 5.2, JMRI Decoder Pro funciona sin problemas y no es necesario utilizar un amplificador de programación con el NCE Power Pro**).

La principal ventaja de poder guardar un conjunto completo de CV en un archivo se manifiesta cuando hay una maqueta con muchos desvíos y múltiples QSDD para moverlos. Después de configurar un decodificador, y sus cuatro desvíos asociados a su entera satisfacción, puede tomar una copia del CV modificados y transferirlos en una sola operación a todos sus otros QSDD.

Configurar esos otros QSDD, cuando se transfieren de la Vía de Programación a la maqueta, consistirá solo en pequeños ajustes en los movimientos del servo para compensar las diferencias entre servos individuales (y quizás desvíos o enlaces mecánicos).

La siguiente captura de pantalla de la aplicación A-Track muestra la ventana Variables de Configuración para el decodificador con todas las CV relevantes configuradas en sus valores predeterminados.

The screenshot shows the 'Item #1 - Addr 0001 - Configuration Variables' window in A-Track. The 'Item Description' is 'QuadServoDecoder46'. The 'Config' tab is active, showing the following settings:

- Item Type: Accessory
- Acc'y Output Addr: 0001
- Extended Address: =====
- Consist Address: =====
- User Identifier #1: 000
- User Identifier #2: 000
- Manufac Version No.: 089
- Manufacturer ID: 013
- Not Allocated: 00
- Edited:
- Not Prog'd:

At the bottom of the window is a table of Configuration Variables (CVs):

| CV | 0x | 1x | 2x | 3x | 4x | 5x | 6x | 7x | 8x | 9x | 10x | 11x | 12x |
|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 000 | 255 | 255 | 000 | 000 | 173 | 006 | 006 | 255 | 255 | 255 | 083 | 000 |
| 1 | 001 | 255 | 255 | 000 | 001 | 000 | 001 | 001 | 255 | 255 | 255 | 068 | 024 |
| 2 | 255 | 255 | 255 | 000 | 000 | 000 | 070 | 070 | 255 | 255 | 255 | 070 | 000 |
| 3 | 255 | 255 | 255 | 000 | 002 | 000 | 110 | 110 | 255 | 255 | 255 | 255 | 255 |
| 4 | 255 | 255 | 255 | 000 | 000 | 000 | 070 | 070 | 255 | 255 | 255 | 255 | 255 |
| 5 | 255 | 255 | 255 | 000 | 003 | 006 | 006 | 255 | 255 | 255 | 255 | 255 | 255 |
| 6 | 106 | 255 | 255 | 000 | 000 | 001 | 001 | 255 | 255 | 255 | 255 | 255 | 255 |
| 7 | 089 | 000 | 255 | 000 | 004 | 070 | 070 | 255 | 255 | 255 | 255 | 255 | 255 |
| 8 | 013 | 000 | 255 | 000 | 000 | 110 | 110 | 255 | 255 | 255 | 255 | 255 | 255 |
| 9 | 000 | 255 | 226 | 000 | 000 | 070 | 070 | 255 | 255 | 255 | 081 | 255 | |

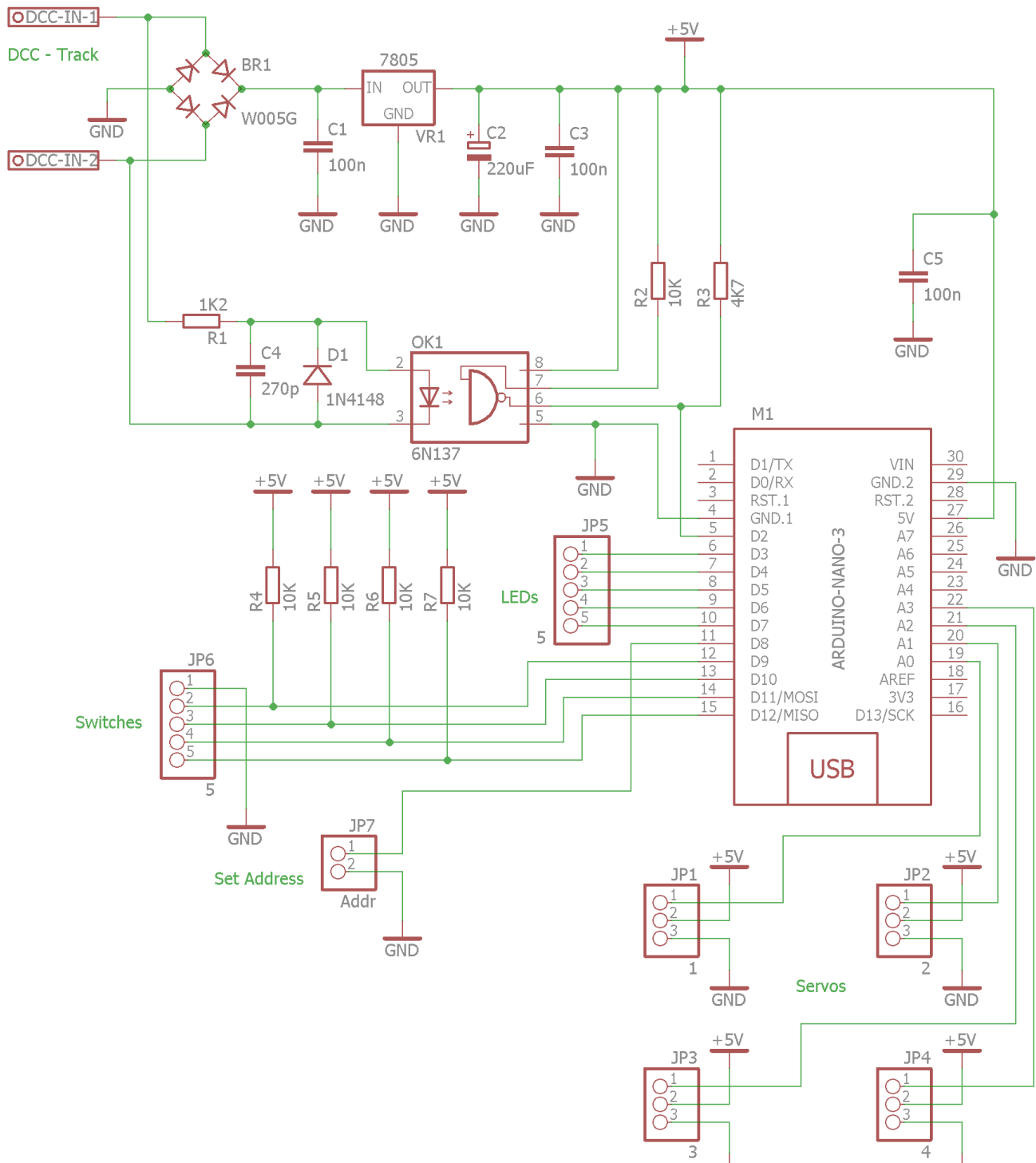
Los ajustes de cualquiera de los valores de CV se pueden hacer simplemente escribiendo un nuevo valor (o conjunto de valores) y luego usar la función Programa de la aplicación para transferirlos al decodificador.

Detalles técnicos

Para quien esté interesado, se muestran los esquemas del circuito del decodificador y del teclado en las páginas siguientes. El circuito de entrada DCC está adaptado a partir del diseño de Wolfgang Kuffer: (<https://mrrwa.org/dcc-decoder-interface/>) y es utilizado por Geoff Bunza como base para varios de sus proyectos.

La señal DCC de la vía en la maqueta (normalmente 14 - 16 voltios CA) se rectifica mediante el puente de diodos BR1 para suministrar alrededor de 14 voltios DC al regulador de voltaje VR1 que, a su vez, suministra +5 voltios al Arduino Nano, sus circuitos asociados y los servos adjuntos. Aunque la corriente normal a través del regulador, con los servos inactivos, es de unos 45 mA, mantiene su disipación de potencia por debajo de 0,5 vatios, que podría aumentar brevemente hasta 10 vatios si se ordena a los cuatro servos moverse simultáneamente. De ahí la necesidad del disipador de calor adjunto al regulador. La señal de entrada DCC también está conectada a la entrada del optoacoplador OK1 a través de la resistencia R1. El condensador C4 filtra cualquier pico de alto voltaje de la vía y el diodo D1 evita que la entrada del diodo optoacoplador sea fatalmente polarizada inversamente por la parte negativa de la señal DCC.

La salida del optoacoplador OK1 es una réplica de la forma de onda DCC, pero a un nivel seguro de +5 voltios, por lo que los paquetes de comando DCC se pueden introducir en la entrada digital D2 del módulo Arduino Nano. Aquí son decodificados por las funciones de la biblioteca NmraDcc y los comandos de accesorios relevantes luego pasado al código de boceto QSDD.

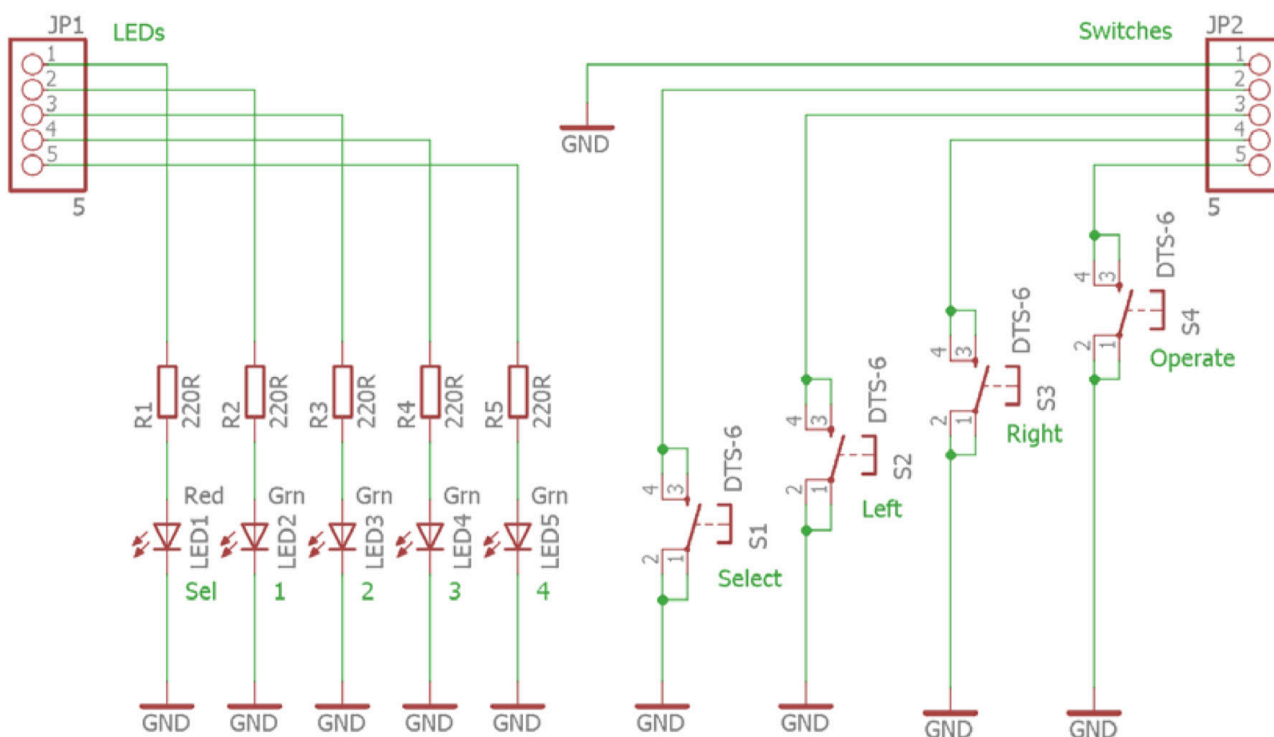


Decodificador Quad Servo DCC - Decoder

El resto del circuito cubre las diversas entradas y salidas del Arduino Nano. Los LED montados en el teclado se controlan desde las salidas D3 - D7 a través del conector JP5 en el decodificador y conector JP1 en el teclado con resistencias R1 - R5 configurando la corriente a través de cada LED a aproximadamente 13 mA.

Los pines D9 - D12 se establecen como entradas que se conectan a los cuatro pulsadores montados en el teclado mediante el conector JP6 del decodificador y el conector JP2 del teclado. Las entradas están normalmente a un nivel ALTO por las resistencias R4 - R7 en el decodificador, pero se llevan a un nivel BAJO nivel (GND ó 0 voltios) siempre que se presiona el botón correspondiente.

El pin D8 se lleva de manera similar a un nivel BAJO siempre que el puente se enchufe en los dos pines *Address* del conector de programación JP7, de dos pines. Éste utiliza un pull-up interno, en lugar de una resistencia externa adicional, para mantener su nivel ALTO cuando el puente no esté instalado. Finalmente, los pines A0 - A3 están configurados como salidas para impulsar uno de los cuatro servos conectados a través de los grupos de tres pines, JP1 - JP4 en la placa decodificadora.



Decodificador Quad Servo DCC - Teclado

Dr. Terry Chamberlain

Terry Chamberlain se introdujo en el modelismo ferroviario casi por accidente en la década de 1990 cuando respondió a una solicitud de algunos modelistas de California para construir un Sistema DCC basado en una computadora personal Atari, y tuvo que construir un diseño simple para demostrar que todo funcionase. Finalmente, el proyecto se convirtió en A-Track, una aplicación de Windows que proporciona soporte informático completo para gama completa de sistemas NCE DCC, con prestaciones similares al decodificador de JMRI Pro y Panel Pro.



Terry es un ingeniero electrónico profesional y pasó la mayor parte de su carrera en la industria de defensa del Reino Unido diseñando y gestionando el desarrollo de grandes sistemas informáticos en tiempo real para el Royal Navy. Ahora que se ha jubilado, todavía espera construir el diseño de tala y minería que ha estado planeando durante años (después de varias visitas a Colorado), pero sigue distrayéndose con nuevos proyectos de informática y electrónica para maquetas de ferrocarriles.