AN-1189 Coin Counter

This application note will describe how to create a piggy bank coin counter with a GreenPAK™. This piggy bank counter will utilize three primary components:

- **SLG46531V**: The GreenPAK IC serves as the interpreter between the sensors and display values. It is also the IC responsible for reducing the power consumption of the whole circuit, by implementing PWM to drive the second component.
- **The CD4026**: The CD4026 is a dedicated IC for driving the 7-segment LED displays. It is quite similar to the CD4033, which can also be used to drive the displays used in this app note. However, it is recommended to use the CD4026 as its Display Enable IN pin will allow us to reduce the power consumption by implementing a PWM.
- **The DC05**: The DC05 is the 7-segment LED display we are going to use. There are several models of display that vary in size and color. Choose the one that appeals most to your taste.

System Operation

The system uses four 7-segment LED displays (DC05), each of which can display a number between 0 and 9. Using four displays, we can achieve a range from 0 to 9999, which is a high enough balance for a typical piggy bank. Figure 1 shows the Pinout of the DC05.
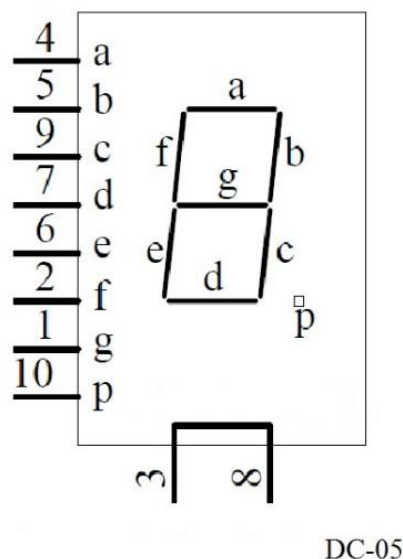

Figure 1. The pinout of the DC05

Every DC05 requires a driver to store and display the value. The CD4026 and CD4033 are excellent options to choose from, and with a range from 5 to 20 volts of operation, we can use them even for big billboards. Both drivers will move through the sequence from 0 to 9 with each pulse sent to CLOCK (Pin 1 in Figure 2).

In this app note we will use the CD4026, because of the possibilities it offers for saving power. Figure 2 shows the Pinout of the CD4026.
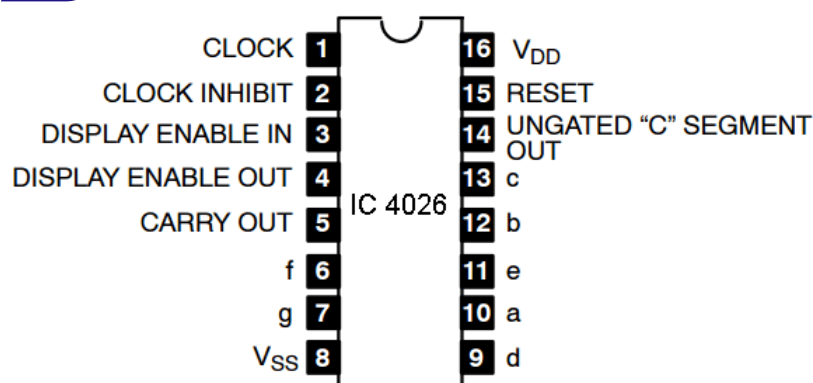
Figure 2. The pinout of the CD4026

Every time the CD4026 receives a pulse on its "CLOCK" input, it increments its internal counter. When the counter value is 9 and the CD4026 is clocked an additional time, it outputs a pulse on "CARRY OUT" and rolls over to 0. This way you can implement a counter from 0-9999 by connecting the "CARRY OUT" signals to the next CD4026 in the array. Our job is to translate the coin values into pulses for the first CD4026, and it will do the rest. Figure 3 shows the basic concept with two sets of CD4026 and DC05.
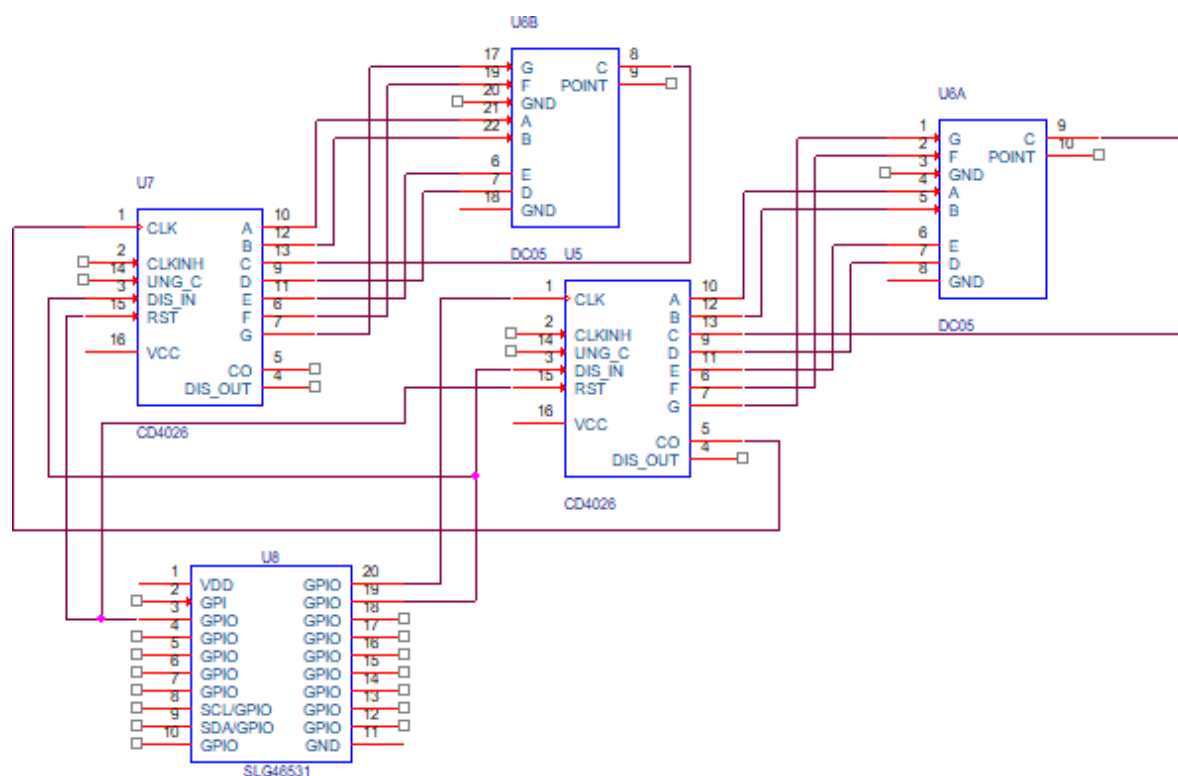

Figure 3. Top Level schematic

The GreenPAK is responsible for recognizing the kind of coin and assigning the correct number of pulses to each one. For this app note, we will use coins valued at 1, 2, 5 and 10 MXN. However, all techniques discussed here can be applied to any currency that uses coins. Now, we have to devise a way to distinguish between different coins. There are several methods to do this, including utilizing the metal composition of the coin and the diameter of the coin. This app note will use the latter method.

Table 1 shows all the diameters of the MXN coins used in this app note, as well as the diameter of US coins for comparison.

Value	Diameter in mm	Currency
1	21	MXN
2	23	MXN
5	25.5	MXN
10	28	MXN
1	19.05	US CENT
5	21.21	US CENT
10	17.91	US CENT
25	24.26	US CENT
50	30.61	US CENT
1	26.49	USD

Table 1. The diameters of the MXN coins

There are several ways to determine the diameter of a coin. For example, we could use a plate with coin-sized holes like in Figure 4. Using an optic sensor, we could signal every time a coin passes through a hole, and send the corresponding value in pulses. This solution is bigger and bulkier than the one we'll use for this app note, but it may be easier to build for a hobbyist.
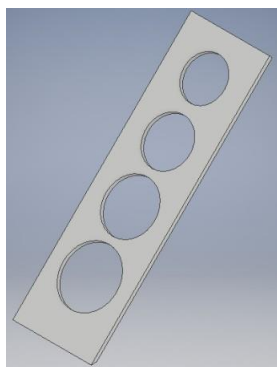

Figure 4. Coin-sized holes in a plate

Our solution will use a mechanism taken out of a broken toy, shown in Figure 5. It would be a relatively simple task to build a replica using wood.

Figure 5. Detachable mechanism

Coins can be inserted in the slot at the left edge of the mechanism in Figure 5. This slot will be forced down by a certain distance based on the diameter of the coin. The metallic piece circled in yellow will be used to signal the size of the coin, and the spring will push the slot back into starting position. This sensor will activate multiple readings each time a coin its inserted; for example, when a 10 MXN coin is inserted, the sensor will briefly touch the values of 1, 2, and 5. We must take this into account in the next part of the design.

GreenPAK Design Implementation

The system works in the following way:

1. The sensor is in the starting position.

2. A coin is inserted.

3. The sensor moves from the smallest diameter to the correct one, based on the diameter of the coin.

4. The spring returns the senor to the initial position.

For example: A 10 MXN coin will displace the sensor from the starting position into the 1 MXN position, then the 2 MXN position, then the 5 MXN position, until finally arriving at the 10 MXN position before returning to the initial position.

To handle this problem, we'll implement a one way ASM inside the GreenPAK, shown in Figure 6.
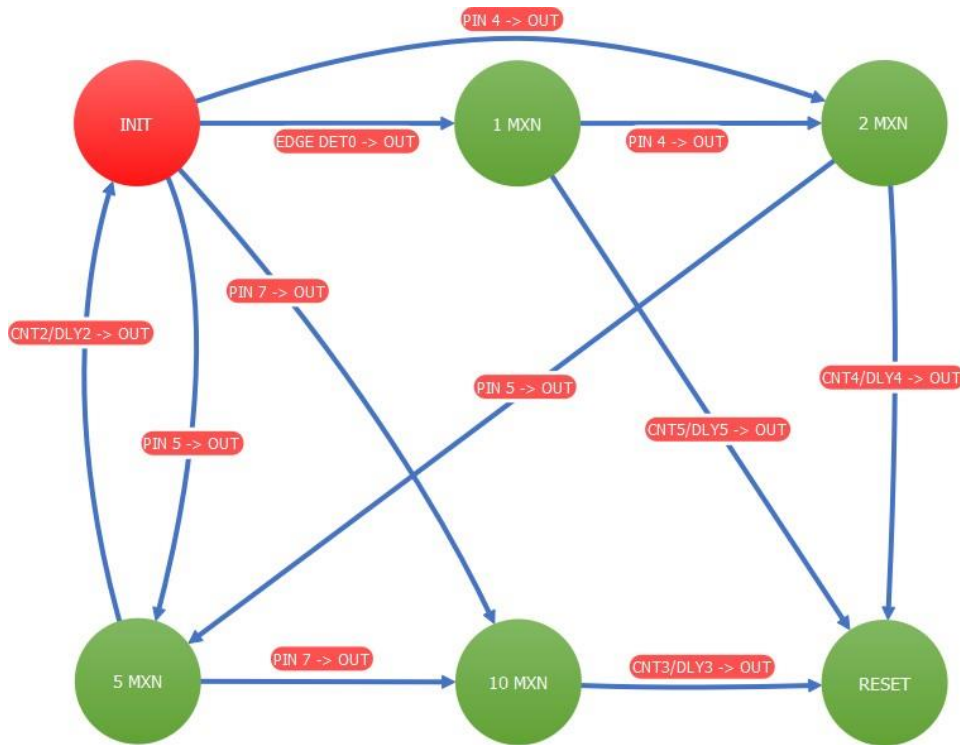

Figure 6. ASM State diagram

Once the sensor is in the starting position, the state of the ASM determines how many pulses the system is going to send.

For the system to send the pulses, three conditions must be met:

1. The system must be in a valid state (1 MXN, 2 MXN, 5 MXN, or 10 MXN).

2. The sensor must be in the starting position.

3. There must be a pulse to be sent.

Counting the pulses is a difficult task, because the counter will output a HIGH when the value is reached, and it will also send a HIGH when the counter is reset. If the counter is not reset, then the output will remain HIGH.

The solution is rather simple, but difficult to find: count to the coin value plus one, and reset the main oscillator with the rising edge of the sensor returning to the starting position. This will create a first pulse that will make the counter of the current state count up to the coin value. Then, add an OR gate to the output into the CLK input (along with the signal from the oscillator) to achieve a reset of the system.
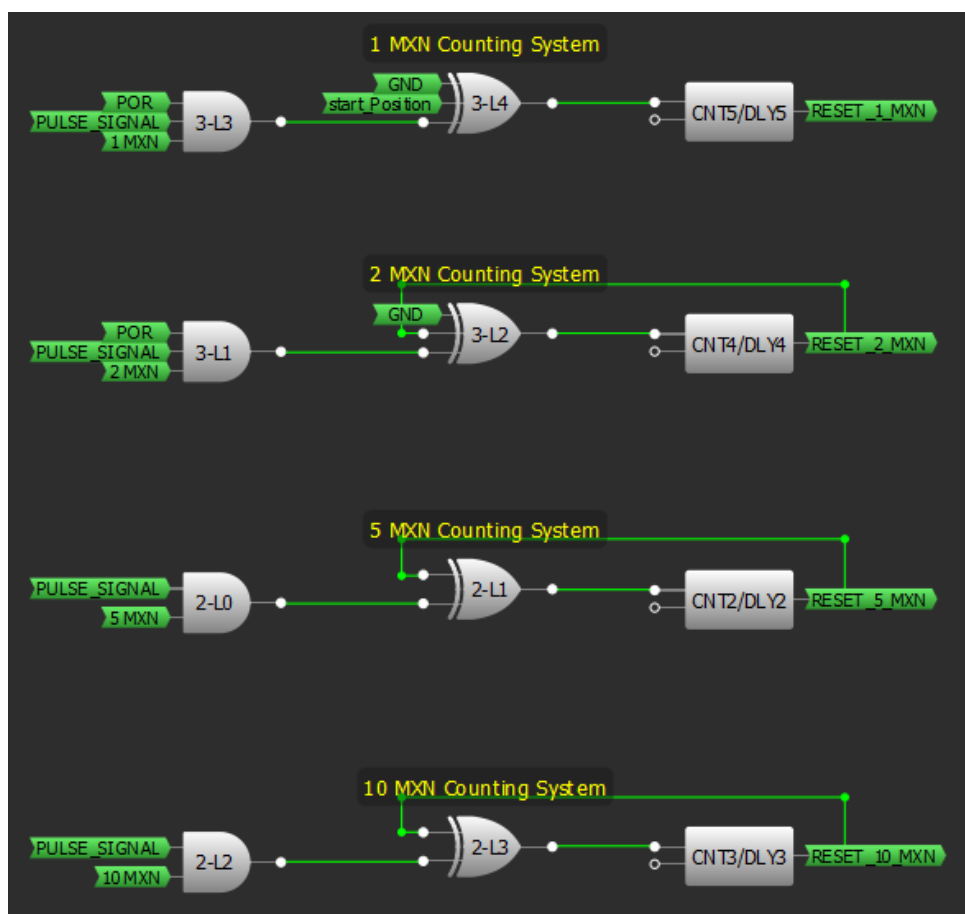
Figure 7 depicts this technique.

Figure 7. MXN Counting System in GreenPAK Designer schematic

After counting to the coin value, the system sends a reset signal back to the ASM to return to INIT. A close look at the ASM is provided in Figure 8.
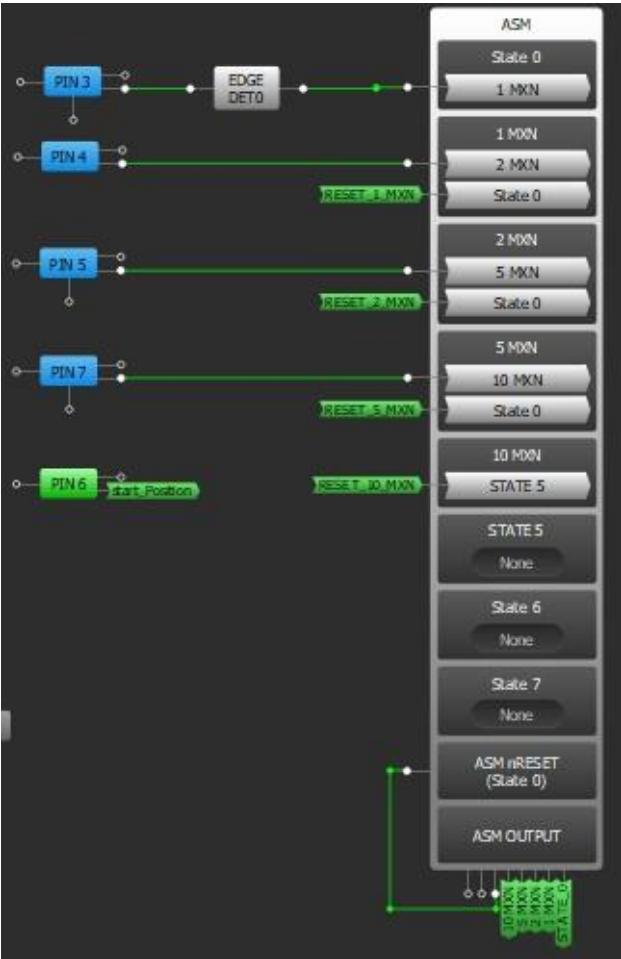
Figure 8. ASM Output in GreenPAK Designer schematic

RESET_10_MXN uses a slightly different system than described above, using an extra state to restart the whole ASM, as there is a limited amount of connections each state can have. The RESET_10_MXN was achieved by going to the RESET state, which was the only state where the ASM's OUT5 was LOW. This successfully returns to the INIT state without any problems.

CNT2, CNT3, CNT 4, and CNT5 share the same parameters, except the value of the counter shown in Figure 9.

Figure 9. Counter Properties

As the CD4026 uses the rising edge of the signal to advance its sequence, this system counts the rising edges values. A low frequency was selected for debugging purposes. Using higher frequencies would be useful and can be done without major problems.

In order to implement this app note in any other currency, simply adjust the counter to the value of the coin plus one.

Using other sensors would make this system far simpler, but production costs would be higher than solving these issues through programming.

Test Results

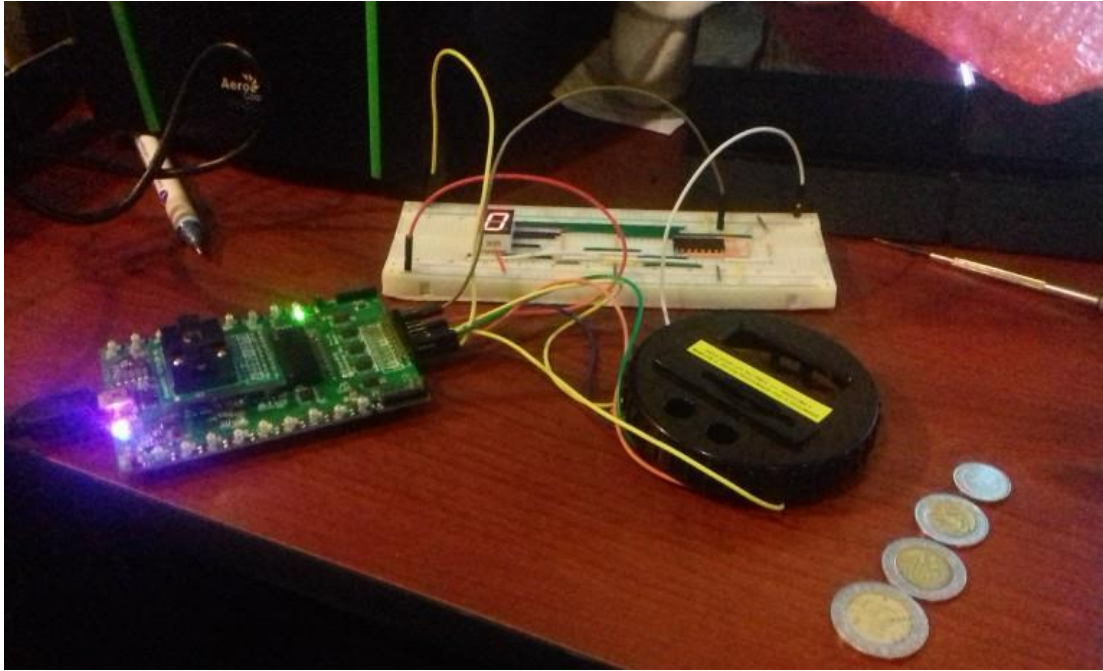The complete project setup is shown in Figure 10.

Figure 10. Entire project setup

Diameters were adjusted to work with different coins, and the denomination can be changed by altering the using the .gp5 file.

Conclusions

Thanks to the GreenPAK product line, it is easy and affordable to develop a system like this piggy bank. The project could be further enhanced by using a PWM signal to drive the CD4026 Display Enable IN. You could also use the GreenPAK to generate a wake/sleep function to lower the system's power consumption. This simple system could be used to control a variety of coin- accepting systems, like vending machines, arcade machines, or coin lockers.

IMPORTANT NOTICE AND DISCLAIMER

RENESAS ELECTRONICS CORPORATION AND ITS SUBSIDIARIES ("RENESAS") PROVIDES TECHNICAL SPECIFICATIONS AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD-PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for developers who are designing with Renesas products. You are solely responsible for (1) selecting the appropriate products for your application, (2) designing, validating, and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. Renesas grants you permission to use these resources only to develop an application that uses Renesas products. Other reproduction or use of these resources is strictly prohibited. No license is granted to any other Renesas intellectual property or to any third-party intellectual property. Renesas disclaims responsibility for, and you will fully indemnify Renesas and its representatives against, any claims, damages, costs, losses, or liabilities arising from your use of these resources. Renesas' products are provided only subject to Renesas' Terms and Conditions of Sale or other applicable terms agreed to in writing. No use of any Renesas resources expands or otherwise alters any applicable warranties or warranty disclaimers for these products.

(Disclaimer Rev.1.01 Jan 2024)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit www.renesas.com/contact-us/.