

Z18 Installation Instructions

This model of time clock is designed to be professionally installed. It is meant to be permanently mounted with the wiring invisibly buried behind the clock. It is not meant for desktop or tabletop use. If you want a desktop or tabletop clock, we strongly recommend the FlexClock Vx510 (tabletop) or FlexClock Z11 (wallmount), as they are much simpler and do not require in-wall wiring.

The Basics

Your clock comes shipped with a metal backplate attached. It also comes with a Torx screwdriver to separate the backplate from the clock. The backplate goes on the wall separately, along with the rubber mat. The rubber mat goes between the backplate and the wall.

The power connector provides a bare red and black wire for 12 volts DC power. Included is a power adapter cord with bare wires. If necessary, extend the power connection with low-voltage wire (such as 18 gauge speaker wire) to a convenient or hidden power outlet. You must observe proper polarity in order to prevent damage to your clock. The power adapter is marked with a notation (such as red tape) to indicate which wire is positive. This must go to the red power wire. Reverse polarity installation will damage the clock. Damage due to improper installation is not covered under the warranty.

Installation - Wires buried in wall

The backplate has a cutout to allow wires to exit into the wall. Simply cut a hole in your wall to coincide with the hole in the backplate. A mounting template is provided for your convenience.

In order for your clock to function, you need to get **power** and an **Ethernet** connection inside the wall at the site of the clock.

Your clock comes with simple drywall anchors. Ideally, the backplate should be screwed firmly into a stud or some type of sturdy backing. If there is no nearby stud but you cut a large enough wiring hole, you may be able to affix a small wooden block on the inside of your wall with screws or glue to provide a stronger level of anchoring.

Installation without wall invasion

If you do not wish to bury the wiring in the wall, you may consider mounting the clock onto a cabinet or furnishing.

Regardless of how you mount the clock, it is vital that you protect the wiring harnesses on the back from cables being tugged out by accident. The harnesses were not designed to withstand forces on the wires,

and the connectors can be easily damaged. If the wires are exposed, please secure them in a way that prevents damage to the connectors if they are inadvertently pulled.

Wiring the clock

At a minimum, you will use two (2) of the wire harnesses to connect your clock: one for power, and one for Ethernet.

The power harness has two wires: red and black. The red connects to 12 volts DC, the black connects to ground.

The Ethernet harness has four wires. For your convenience, an RJ45 jack is attached to the harness.

Three more harnesses are included. One is 7 pins, one is 8 pins, and the last one is 10 pins. The remaining wiring harnesses are necessary only if you wish to use any of the following features provided by the clock:

- External Bell Ringer
- External Proximity Card Reader
- External Fingerprint Reader
- External Barcode Reader

If you attach these harnesses, please cut and separately tape off any unused wires so as to prevent damage to the clock through short circuits.

Hanging the clock on the mounting plate

The clock hangs on the mounting plate in a somewhat reversed fashion. Rather than hanging on the mounting posts and sliding downward, the clock must be installed on the backplate's mounting posts and then pushed **upward** to bring the mounting plate's screw threads near the hole in the bottom of the clock. Install the Torx screw to mount the clock.

Bell Ringer Installation Instructions

Your time clock has a feature that allows it to ring an external bell on a schedule. The time clock rings the bell by connecting two contacts on an internal relay marked "**COM**" and "**NO**". You may wire this relay as though it were a switch controlling access to the bell.

The bell relay can control virtually any device, but to avoid damage to the time clock, we strongly recommend that you not connect any power source exceeding 24 volts, or any loads exceeding 1 amp. If you need to control a device that requires higher voltage or amperage, then you should use an intermediate larger relay (such as the type used in car stereo/car alarm installations) to control the larger load.

Don't connect anything to the pins on the time clock marked "BELL". This connector is meant for a doorbell feature, where the contacts close whenever the doorbell key (lower left key on the keypad is pressed), and not on the bell schedule as you may have intended.

Connecting the bell

Using an external power supply

(diagram)

Using the clock's 12-volt power supply

(diagram)

Using an intermediate 12-volt relay

(diagram)

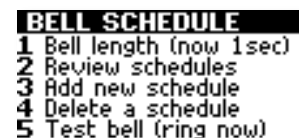
Setting up the bell schedule

To access the bell schedule menu, press the MENU key twice, and then enter the password 74642355 (spells "RINGBELL" on a phone keypad).

From the bell schedule menu, you can set the times and days when the bell should ring. You can either set schedules by day of the week (e.g. Monday thru Friday), or by specific days (e.g. December 25), or a combination of both.

Bell schedule changes take effect immediately. They are not transmitted from clock to clock, so they affect only the clock being programmed.

If the power is turned off when a scheduled bell event occurs, the bell will *not* ring. There will be no "make-up" ring even if the power comes back on later - unless the power comes on within 60 seconds of the time it is supposed to ring.



BELL SCHEDULE
1 Bell length (now 1sec)
2 Review schedules
3 Add new schedule
4 Delete a schedule
5 Test bell (ring now)



Add Schedule
Will this schedule be
daily/weekly, or for specific
date(s)?
1 Daily or weekly
2 Specific date(s)

Universal Prox Installation Instructions

Your time clock has a feature that lets it connect to any Wiegand-compatible access control device, such as a proximity card reader. In most cases, the time clock can also decode the card number from the Wiegand signal. **These instructions do not apply if you're using only the built-in proximity reader.**

Wiring the proximity card reader

The connection between the time clock and the proximity card reader consists of three (3) wires: black, green, and white. These are known as ground, Data 0, and Data 1, respectively.

The Wiegand data pins do not provide power, so the proximity card reader will need external power. If the reader can run on 12 volts at less than 1.0 amp, then you can connect it to the red +12 Power Out on the back of your time clock. If it needs any other voltage, then you need to power your external reader separately according to its manufacturer's instructions. If you use external power, the GND (ground) connection must still be connected between the reader and the time clock in order for the reader to function.

Your proximity reader should have a Wiegand connection consisting of three wires: DATA 0 (green wire), DATA 1 (white wire), and GROUND (black wire). These connect to INWD0, INWD1, and GND on the 7-pin harness, using wires of the same color.

Avoiding Interference from Z18's Built-In Reader

The Z18 has a built-in proximity card reader. This reader can cause cancellation of your external proximity card reader's signal if their antennas are too close. This problem only affects the Z18, as the L1 doesn't have a built-in proximity card reader.

We recommend you locate the external reader at least 12 inches away from the Z18. You might want to test your installation before mounting it to see if interference will be a potential issue.

You could also overcome this limitation by opening the Z18 and disconnecting the internal card reader's antenna. The antenna is a loop of varnished copper "magnet wire" (possibly red in color) that winds around the front of the clock. It connects to the clock via two wires plugged into a socket labeled "ANT" (for *antenna*). Carefully disconnect the antenna connector from the circuit board. To preserve the delicate antenna wire and your warranty, and to avoid ripping the socket from the board, it is best if you pry the connector loose with a small screwdriver or blade, instead of trying to pull it out with your fingers.



Figure 1 - Antenna connector inside Z18

Configuring the clock for Universal Prox

Once the wiring is completed, the clock needs to be configured to accept and decode the Wiegand signal.

To access the Wiegand menu, press MENU twice, and then type the password 9434263 (the numbers spell WIEGAND on a telephone).

Wiegand "Bits"

At the very minimum, you need to know the number of "bits" (count of binary ones and zeroes) your proximity card system sends for each card swipe. The most common number of bits is 26, but many proximity card systems use more than 26 bits. If you do not know the number of bits, then you can do a simple trial-and-error session to find out the number of bits.

If you go to the test screen, you will always see at least **something** when any card is swiped, so long as the bit setting isn't **too high**. Keep increasing the number of bits until you find the **greatest** number that consistently responds to a card swipe in some way (even if the response is just an error). As soon as the test screen starts missing card swipes, decrease the number of bits by 1, and you've now found the correct number of bits.

Getting the right bit setting is important. Even though the system will appear to read cards just fine at a bit setting that is too low, it might be missing part of card number. This will result in multiple cards reading as the same card number when swiped, making the system unable to distinguish all the cards from each other and causing cards to appear as duplicates.

If it turns out that you have a 26-bit system, your system might use the common industry standard. In that case, you should consider going to the Presets menu and selecting "Generic 26-bit Wiegand". This sets the *interpretation* and *validation* options to match the industry standard. If that doesn't work, try blanking out the Interpretation, Parity, and Facility Validation options to enable the "hash" mode described later.

Wiegand "Interpretation"

Interpretation is the process of translating the ones and zeroes into a readable card number. There are two ways interpretation can be done.

The first way attempts to derive the actual card number printed on the face of the proximity card. In order to do that, you need to enter a short "script" that tells the clock how to do this. An example script might look like this: "S9N16". It means, "Skip 9 bits, and then read the next 16 bits as the card number". This is the script used for generic 26-bit Wiegand.

If the first way is unsuccessful, the alternate "second way" gives a second-best solution that is guaranteed to work, but it cannot determine the actual card number. Instead, a semi-random 8-digit number is generated for each card, using a predictable "hash" function. See the section on Wiegand "Hash" mode for more information.

To enter interpretation scripts, please note you will need to enter alphabetic characters on the clock keypad. You can enter alphabetic characters first by typing a number, and then pressing the F3 (ABC/Alpha) key to switch those numbers to become letters. The number mapping is similar to a telephone, so the number 2 can be switched to A,B,C, the number 3 can become D,E,F, etc.

All of the commands are based around an imaginary "pointer" that can be used to scan the bits. At the beginning of the script, the "pointer" is always pointing at the first bit. Commands you execute always look at bits starting from the pointer, and in doing so, the pointer is moved to the right.

The commands are as follows. After each command, you need to put a number to tell how many bits should be acted upon.

* S - Skip bits. (Moves an imaginary "pointer" past the number of bits you specify)

* N - Accept bits into the card number, starting from the imaginary "pointer".

The following example shows how interpretation might work using generic 26-bit Wiegand.

When Generic 26-bit Wiegand is used, the bits are allocated as follows: one parity bit, then 8 facility bits, then 16 card number bits, then another parity bit.

P F F F F F F F C C C C C C C C C C C C C C C P

↑

Notice the arrow points at the first bit. This is where our "pointer" always starts at the beginning of each script.

We will use the following card as an example, which has a facility code of 13 and a card number of 37516. You can use the "scientific" mode of Windows Calculator to convert decimal numbers to binary and back.

13 = 1101 in binary (or 00001101, if padded to occupy 8 bits)

37516 = 1001001010001100 in binary

This card yields the following 26 bits, shown with the card number underlined, and the pointer:

1 0 0 0 0 1 1 0 1 1 0 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1

↑

During interpretation, we are only interested in the card number. (If there is a need to validate the facility code, a Facility Validation Script can be used, described later).

The first thing we want to do is skip the pointer so it's pointing at the card number. There are 9 bits that must be skipped, hence the command is **S9**. After this command finishes, the pointer has been moved ahead:



Figure 2 - A card with a 26-bit Wiegand number, with its code shown both as a single number (889484) as well as broken out into parts (facility 13, card 37516).

1 0 0 0 0 1 1 0 1 1 0 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1



Next step is to read the 16 "C" bits as the card number. The command **N16** does that.

1 0 0 0 0 1 1 0 1 1 0 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1 Card#: 1001001010001100 = 37516



After N16 finishes, the pointer moves to the very last bit, but the card number has been "copied" into a special buffer that will ultimately be converted back to the decimal card number of 37516.

Alternately, let's suppose that on the example card, you wanted to read all of the card number bits without separating the facility from the card number. The other number on the card (889484) is the result of interpreting *all* the bits together with the script **S1N24**. The script skips the first parity bit, and then reads all of the bits together (except the final parity bit) to form the card number:

1 0 0 0 0 1 1 0 1 1 0 0 1 0 0 1 0 1 0 0 0 1 1 0 0 1 Card#: 000011011001001010001100 = 889484

Wiegand Facility and Parity Validation

Your clock can be set up for optional parity and/or facility validation. These are two more similar "scripts" that can be used to reject invalid cards. Validation settings are entirely optional. If you don't set them up, the clock will simply accept any card swipe that provides the minimum number of bits.

The S (skip) and B (move pointer) commands work exactly the same. Three new commands are provided to implement the validation of "even parity", "odd parity", and enforcing matches of groups of bits.

- E - Assert even parity over a given number of bits, starting with the imaginary pointer. Example, "E13" requires that there be an even number of 1's among the first 13 bits starting with the one being pointed to. If the number of 1's is odd, then the validation fails. If the validation succeeds, the pointer moves ahead the number of bits just tested (so it will be pointing at the first untested bit following the ones that were just tested).
- O - Assert odd parity. Works just like E, except requires odd number of 1's, not even.
- M - Match a pattern. Requires that the bits being pointed at must match a given pattern. This is useful because many cards may have a "facility code" that isn't part of the card number, but which should nevertheless be validated, and that is what this command is for. Matching is always against the numbers 0 and 1, but the number 9 can be used as a placeholder wildcard. Example: "M10090001" tests 8 bits starting from the pointer, and due to the wildcard, they must be either "10000001" or "10010001". If match is successful, the pointer moves to the bit just after the last bit that was tested.

If you do not wish to use Parity/Facility validation or you do not know the layout of your access control system's existing cards, then you should leave the validation settings blank.

Parity Validation Example

The following example shows how parity validation might work using generic 26-bit Wiegand.

When Generic 26-bit Wiegand is used, there is a rule that the first 13 bits must have "even" parity, and the last 13 bits must have "odd" parity. Having "even" or "odd" parity simply means that the total count of 1's must be either an even or odd number. If a number is received that violates the parity rule, it is considered invalid (possibly an indication of some sort of malfunction) and is supposed to be rejected.

Let's borrow the card number from the example above. This would mean that in the underlined portion, there must be an *even* number of ones, and in the remaining portion, there must be an *odd* number of ones. The parity bits (very first and last bits) are meant to be set to whatever is necessary to make the parity rules work.

10000110110010010100011001

↑

The 1's can be counted by hand. There are 6 underlined 1's, and since 6 is an even number, even parity is good. There are 5 double-underlined 1's, and 5 is an odd number, so the odd parity is good.

The command **E13** verifies the even parity. After it executes, the pointer gets moved ahead:

10000110110010010100011001

↑

Finally, the command **O13** verifies the odd parity, starting from where the pointer was left.

If at any time a parity check fails, then the validation script stops running. If you're in the Card Test screen, you will be told that the parity check failed - otherwise, if at the regular Clock In/Out prompts, the card swipe would simply be rejected.

In summary, the script required to perform this validation would be entered as **E13013**. It is vitally important to enter the "O" (as in "ODD") as a letter, rather than the number zero.

If your access control system uses something other than 26-bit Wiegand, it might have different parity rules. If you do not know the parity rules for your system, you can disable it altogether simply by leaving the parity script blank. If the script is blank, all swipes will be considered to pass the parity check.

Facility Validation Example

Facility validation is optional and is not enabled by default. You can enable it if you know that all valid cards bear the same facility code, in order to reject invalid ones. The following example shows how facility validation might work using generic 26-bit Wiegand.

When Generic 26-bit Wiegand is used, a common way to allocate the bits is as follows: one parity bit, then 8 facility bits, then 16 card number bits, then another parity bit.

P F F F F F F F C C C C C C C C C C C C C C C P

↑

Notice the arrow points at the first bit. This is where our "imaginary pointer" always starts at the beginning of each script. Now, suppose we wish to accept only cards with a facility code of thirteen (00001101 in binary). The first thing we will want to do is move the pointer off the P and onto the first bit of F. In order to do that, the command is **S1** (which stands for "skip 1").

P F F F F F F F C C C C C C C C C C C C C C C P

↑

Now that the pointer points at the facility code, we can use the command **M00001101** to require that the next 8 bits match our desired pattern. If the pattern does not match, then the validation script immediately stops and the validation is considered to have failed. If the pattern were to match, then the pointer would move ahead as follows, where additional validation steps (if any) could continue.

P F F F F F F F C C C C C C C C C C C C C C C P

↑

So, although generic 26-bit Wiegand does not require the use of any sort of facility validation script, the script in the above example would have been entered as follows: **S1M00001101**. This script simply enforces a facility code of 00001101 and rejects any card swipes that don't have it.

If any step of the validation script fails, then the validation script stops running. If you're in the Card Test screen, you will be told that the facility check failed. If a failure occurs at the regular Clock In/Out prompts, the card swipe would simply be ignored.

"Hash" Mode

Hash mode is useful when you have no available information on the format of the card or how the bits form a card number. This is common with proprietary access control systems. They still use Wiegand, but they use a non-standard bit count, and/or they assign the meaning of the bits themselves in unusual ways.

Instead of trying to interpret the card number, the clock aims simply to distinguish cards from one other. For each unique card, it generates a placeholder card number you can use in the timekeeping system.

Hash mode is easy to set up. The only thing you must know is the number of bits, which you can quickly determine through trial and error (please use the highest number that gives reliable responses to swipes on the card test screen). Hash mode is automatically enabled if you set all the script settings to **blanks**.

When hash mode is active, an 8-digit card number is automatically generated for each unique card that is swiped, based on the bits received, using a mathematical "hashing" algorithm. The numbers might appear to be random and will have no apparent relationship to any numbers printed on the physical card - but so long as the settings are identical, the same number is guaranteed to appear each time the same card is swiped.

In order to set up employees when Hash Mode is active, each card must be swiped with the reader in Test Mode to determine the unique 8-digit number that will be generated from that card.

Advanced Wiegand Interpretation/Validation Commands

There are a few more commands that can be used in scripts. These are useful if the card number or parity is broken into several parts throughout the Wiegand message, as is possible on some proprietary access control systems.

- B - Moves the pointer to a point at a fixed position in the bit stream, where 0 is the first bit. This allows the pointer to be moved arbitrarily, including backwards if necessary. Example: B3 moves the pointer to the fourth bit in the message.
- X - This captures bits to become part of the next parity calculation. It is useful if parity must be calculated over a non-contiguous block. Example: X3S3E10 moves the pointer a total of 16 bits, but checks for even parity over 13 bits: the first three and the last ten. The three bits that were skipped with the S command don't become part of the calculation.
- H - This activates hash mode, but allows you to hash only the bits of the Wiegand message you select, instead of the entire message. To use it, use the N command to capture the bits you want to hash, just as though you were capturing a card number. Then put H at the end. Please note that leading zeroes affect hash output. The hash of "00001010" produces a different card number than hashing "1010".
- You can use the N command more than once in the same Interpretation script to capture additional chunks of bits. Each successive time you use the N command, the additional bits are tacked on the end of whatever bits you may have already read. This allows you to read a card number that may be broken into chunks throughout the Wiegand message. Whichever bits you read *first* will be considered the "most significant" bits.

Wiegand Forwarding (Optional)

On the back of the clock, there is a connection for Wiegand Output. This can be connected to your existing access control system if you have one, if you would like to use your external prox reader as a door opener in addition to timekeeping. Wiegand Output is a three-wire connection; you must connect Data 0, Data 1, and Ground (green, white, and black) to your access control device.

Whenever the time clock receives a card swipe it was not expecting (e.g. the IN or OUT button had not just been pressed), the clock automatically forwards the command over the Wiegand Output pins. On the other hand, if the card swipe was expected, either due to IN/OUT being pressed or card reader test mode being active, the swipe is captured and not forwarded.

The Wiegand Out pins are simple TTL logic pins from the clock's CPU board. The signals are not very strong, so the wiring must be limited to a maximum of 40 feet (or possibly less, if the wiring is subjected to electromagnetic interference). If you wish to use the Wiegand Out feature and you need a greater range, you should use a Wiegand signal repeater.

PrintReader Installation Instructions

The Z18 time clock can optionally connect to a PrintReader, which is an available external fingerprint reader accessory that can be mounted on the wall alongside the Z18.

When using a PrintReader with the Z18, the time clock will exchange prints only with other clocks that also use the external PrintReader. The PrintReader contains its own flash memory and processor, and performs fingerprint matching internally. The fingerprint template files generated by the external PrintReader are not compatible with those generated by the L-Series or Z-Series FlexClocks with integrated fingerprint readers.

The PrintReader uses the two RS232 wires for data exchange. The PrintReader can operate on 12 volts DC, so it can easily take its power from the Z18 time clock.

The PrintReader has a round cutout in the backplate that allows wiring to be passed into the wall. Ideally, when a PrintReader is used with the Z18, all of the wiring should be buried in the wall.

The PrintReader has an 8-pin harness on the back, though when it is used with the Z18, only 4 of the pins will actually be connected.

Pin 1 on PrintReader (RED) is +12V, can run from +12V POWER OUT on Z18

Pin 2 on PrintReader (YELLOW) should go to RXD (grey wire) on Z18.

Pin 3 on PrintReader (GREEN or BLUE) should go to TXD (purple wire) on Z18.

Pin 8 on PrintReader (BLACK) is ground, and should go to GND on Z18.

All other pins on PrintReader MUST be left disconnected.

The PrintReader makes three short beeps and flashes its internal LEDs to indicate it is powered up. Once the PrintReader is connected, it must be activated on the Z18 by pressing the MENU key twice, and entering the password 346437 (spells "FINGER").

Barcode Reader Installation Instructions

Your Z18 can connect to an external barcode reader. The barcode reader can be used both as a card number entry method, or optionally as a method to enter the answers to custom clock prompts.

The Z18 is only compatible with barcode readers that use an RS232 interface. Also, you will likely need to purchase your own connector shell that converts the barcode reader's DB9 port into bare wires for connection to the Z18. The barcode reader must supply its own power and most likely has its own power adapter - barcode reader power cannot be taken from the clock.

If you use a barcode reader, you cannot connect a fingerprint reader, as they share the same port.

Wiring the barcode reader

Only two wires out of the 9-pin connector will actually connect to the Z18. The barcode reader connects to the GND and RXD pins on the 8-pin harness. The wire colors are black and grey respectively. The black wire connects to pin 5 of the barcode reader's DB9 port, and the grey wire goes to pin 2.

Most DB9 connectors have very small numbers next to each of the 9 pins identifying the pin number, visible with a magnifying glass, to help identify the pins.

Configuring the barcode reader

You will need to consult the barcode reader's manual to find out the **baud rate** being used by the reader, so you can enter it into the Z18. You will also to configure the reader to send the scanned number plus carriage return - which is typically done by scanning some special barcodes printed inside the manual.

A typical default baud rate used by barcode scanners on the market today is 9600. Other common settings are 1200, 2400, or 19200.

Configuring the Z18

To get to the RS232/barcode configuration screen, hit MENU twice, and enter the code 77232 (spells RS232 on a telephone).

Enter the baud rate used by the scanner. (If this setting is not correct, the scanner **will not work**).

Enter the list of locations where you want the scanner to be usable. Typically this will just be for PIN number entry (if using it as a card reader). "Prompt" settings should only be enabled if you are using the reader for data collection involving custom prompts.