Scaling using y=mx+b – a "no-math" shortcut approach - (part 1 of 2)

foreword by Ron Beaufort: I have previously posted much of the material contained in this paper on the www.PLCs.net "Live Question and Answer" forum at http://www.plctalk.net/qanda/showthread.php?t=15069 ... this was in response to a series of posts requesting help with the subject of "scaling" ... anyone who is interested in following the full context of the questions and answers of the original threads should be able to find them at the following links:

http://www.plctalk.net/qanda/showthread.php?p=99082&postcount=1

http://www.software.rockwell.com/forum/rslogix/messageview.cfm?catid=13&threadid=9174#31731

since my original postings, several readers have requested that I make the text and illustrations easier to access by providing a single continuous PDF file of the material ... this document is the result of those requests ... I've reformatted some parts of the text and added more detail in certain places to make the ideas easier to follow ... most of the forum's original conversational tone has been left in place ...

while I believe that the information included here is correct, it is offered free of charge with no warranty of any kind ... it may be copied and distributed as long as there is no commercial interest involved and as long as the material is copied and transferred in its entirety - including this notice and all notices of copyright ...

I hope that this material proves helpful ... corrections, comments, and suggestions from readers are always welcome ...

someone recently posted a couple of forum requests which went something like this:

I have an SLC-5/02 processor and a 1746-NI4 analog input module. I want to know how to set up the SCL function to read a 0 to 10 volt signal and scale it in my program so that I can display 0 to 3000 psi on my PanelView. I need to know how the math behind this works step by step. I have no experience with analog inputs – this will be my first. I have read the help menu about the SCL instruction and I am lost at obtaining the values I need for Rate, Offset, etc. Scaled max-scaled min and divided by input max-input min. Am I anywhere close? I would appreciate any help anyone could offer.

this got me to thinking about all of the people that I've run into over the years who have needed help with this particular topic ... (and that once included myself) ... what follows is my attempt to cover the subject in enough detail so that anyone can "start from scratch" and end up feeling comfortable with the material and how to use it ...

I've decided to attack this particular subject in reverse ... specifically, I'm not going to start out with an introduction and then work through all of the underlying math involved – and then finally end up with the "cherry on top" shortcut that I personally use ... instead I'm going to get right to the punch line and share the shortcut at the very beginning ...

notice: the author has no connection whatever with the Texas Instruments Company and will receive no compensation, financial or otherwise, from the reader's purchase of any recommended calculators

that having been said, you need to get your hands on a Texas Instruments TI-36X calculator ... they sell for about $19.95 at Wal-Mart ... I'm about to show you a "no math" shortcut that I just know you're going to love ... and while I'm sure that other calculators will do the same type magic, I can give you the exact keystrokes for the TI-36X ... and incidentally, if they're sold out of the TI-36X, do NOT get the TI-30 model ... it's not magic ...

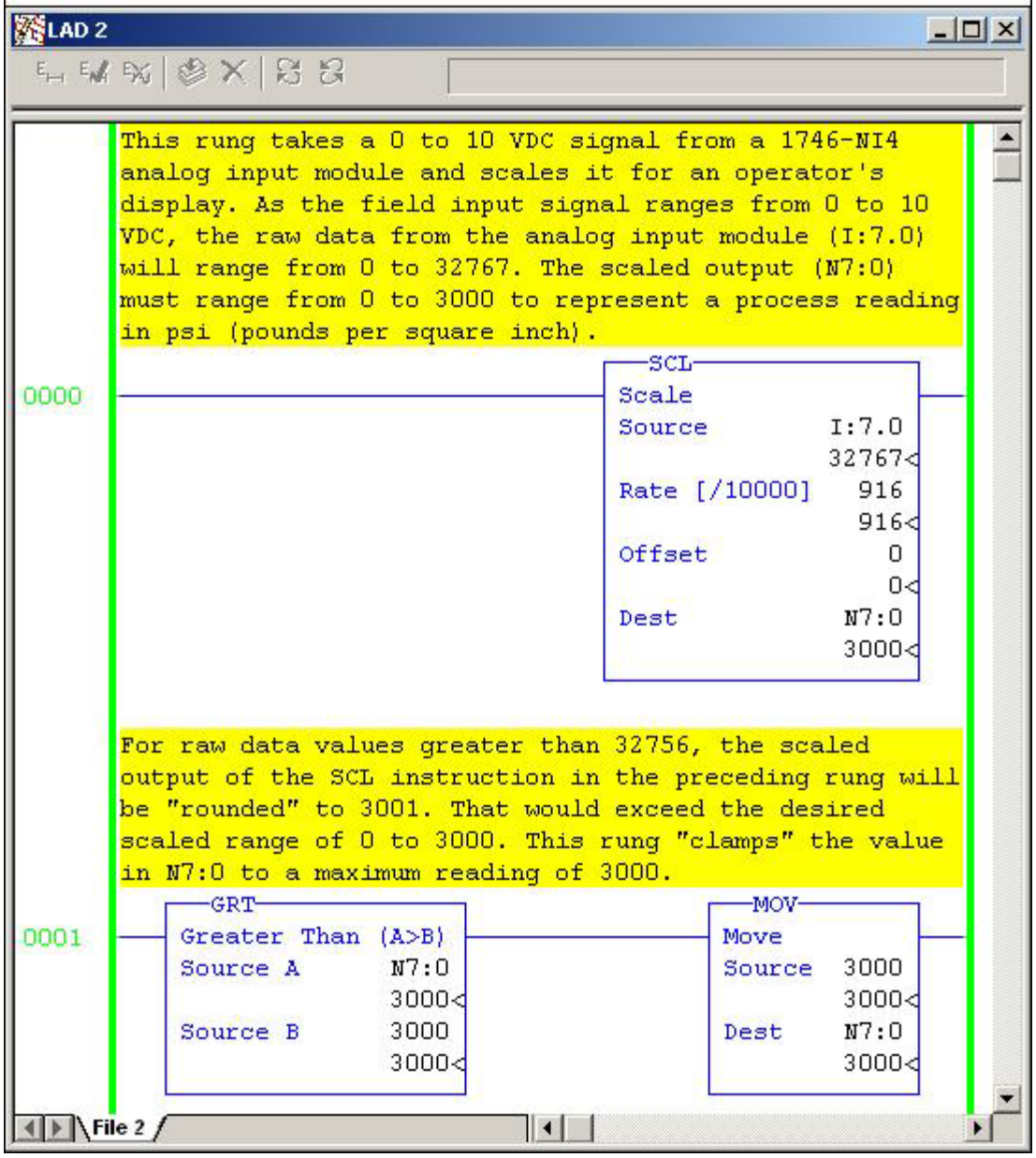
before we use the TI-36X, we need to talk about two of its keys ... the first one is marked with an "X" and a "Y" separated by two little triangles ... (you can see a picture of it in a figure coming up soon) ... I personally call this the "flip/flop" key ... basically it swaps the X and the Y registers – but you don't even need to know that ... the other key is marked with the Greek letter "sigma" and a plus sign ... (sigma looks like a sloppy capital "E") ... I personally call this one the "sum" key ... you're welcome to call these two keys anything you like ... just be sure that you press the right one when the time comes ...

and now let's talk about the SCL (Scale Data) instruction that our original poster needs to use in his program ... note that he's using an SLC-5/02 processor which does not support floating point math ... and unfortunately the easier-to-use SCP (Scale with Parameters) instruction is also not available in his processor ... in simplest terms, with the hardware that he has available, our friend will probably find an SCL the easiest way to get this scaling job done ...

our poster is working with the very common 1746-NI4 analog input module ... for the 0 to 10 VDC input signal that he mentioned, the NI4 will give us raw input data values which range from 0 to 32767 ... and luckily he told us about the 0 to 3000 range that he needs for his scaled output (in psi) ...

Figure 1 shows a very common programming approach which should satisfy our poster's basic scaling requirements ...

Figure 1 - Scaling with an SCL instruction



notice that the SCL's "Rate" entry has been set for 916 ... the "Offset" entry has been set for 0 ... and this brings us to the big question for today: "so how do you come up with those particular numbers?" ... now let's see exactly (keystroke by keystroke) how I used the TI-36X to calculate the values required to scale the signal for this example ...

so the stage is set ... now let's fire up the TI-36X and work through our poster's example problem ... Figure 2 shows the keystrokes of my shortcut method ...

Figure 2 - Using TI-36 Calculator "Stat2" Mode for Scaling

[ON/AC]	turn on and clear calculator
[3rd] [STAT2]	select 2-variable statistics mode
0	value of Input MIN
[X⇆Y]	get ready for next value
0	value of Scaled MIN
[Σ+]	store the MIN values
32767	value of Input MAX
[X⇆Y]	get ready for next value
3000	value of Scaled MAX
[Σ+]	store the MAX values
[2nd] [SLP]	display "Slope" or "Rate" 0.091555528 (answer)
[2nd] [ITC]	display "Intercept" or "Offset" 0 (answer)

pressing [ON/AC] clears out the calculator ... using the [3rd] [STAT2] key combination puts the calculator into the “2-variable statistics” mode ... statistics for scaling? ... oh, yeah ... just hide in the bushes and watch ...

0 ... [flip/flop] ... 0 ... [sum] ...

32767 ... [flip/flop] ... 3000 ... [sum] ...

and that’s all there is to setting up the problem ... now I just need to ask the calculator for the “Rate” and the “Offset” values and we’re just about done ...

the [2nd] [SLP] key combination tells the calculator that I want to know the “slope of the line” ... (we’ll soon cover all of these terms in gruesome detail) ... and the calculator displays 0.091555528 ... this is the “Rate” entry for my SCL - except for one little problem which we’ll cover in just a minute ... now on with the calculator shortcut ...

the [2nd] [ITC] key combination tells the calculator that I want to know the “intercept” ... and the calculator displays 0 ... this is the “Offset” entry for my SCL ...

now notice that so far all I’ve done is crank in four values to tell the calculator the range of my input signal and the range of my desired scaled values ... then I punched a few more keys and got the values that I need to set up my SCL ... NO MATH! ... no multiplication ... no addition ... no division ... no subtraction ... and believe me there’s a lot more “good stuff” to come ... but now back to that pesky little problem with the “Rate” entry that I mentioned earlier ...

remember that the SLC-5/02 processor that our friend is working with won’t handle floating point math ... specifically, no decimal point values are allowed ... so how do we enter the 0.09155 that we need for our “Rate” entry? ... now we’ve come to the reason that the SCL “Rate” entry uses the “/10000” system that confuses so many people ... specifically, it’s a “work around” for the SLC-5/02’s “no decimal point” limitations ... so we take our original 0.09155 value and multiply it by 10000 and that gives us 915.5 and change ... oops, we still have a decimal point ... so next we round off to 916 - and that’s as close as we can come to a perfect answer while using the hardware that we’re working with ... now each time the processor executes the SCL instruction, it will automatically divide the 916 “Rate” entry by 10000 ... (specifically the “/10000” means “divide by 10000”) ... the processor will come up with 0.0916 for a “Rate” value to use for its internal calculations ... secret handshake: the SCL is a “work around” which allows us to get reasonable data resolution with a processor that can’t support floating point numbers ...

quick summary ... we had an input signal which ranged from 0 to 32767 ... we wanted a scaled value which ranged from 0 to 3000 ... we had to use an SCL to do the scaling conversion ... we needed a value for the “Rate” and a value for the “Offset” in order to program the SCL ... I whipped out my trusty TI-36X calculator and cranked in four numbers ... then a few more keystrokes and I provided the “Rate” and the “Offset” values that we needed ... with NO MATH! ... now in the neighborhood that I grew up in, that’s considered some pretty powerful kung-fu ...

on the downside, I’ve been throwing around a lot of weird terms and giving little or no explanation of what they mean ... things like “Rate” ... “Offset” ... “Slope” ... “Intercept” ... and there’s still more weirdness to come before we finish skinning this “scaling” cat ...

next we’re going to tackle the “equation of a straight line” which is usually written “y=mx+b” ... this is a VERY handy math tool and in my personal opinion it’s well worth the time it takes to gain a full understanding of it ... and the good news is that once we understand how it works, the TI-36X calculator will make using it practically painless ...

here’s a little bit of history on how I was introduced to “y=mx+b” ... about 15 years ago I went back to school (technical college) as an adult with bad knees ... a nice fat man was my very first math instructor ... one night he announced that we were about to study the “formula for a straight line” ... “this will be a piece of cake”, I figured ... how hard can a simple straight line be? ... well, I’ll let you decide that for yourself after we’re finished here ... but first I’m going to do something

for you that the nice fat man didn't do for me ... I'm going to try to give you the "big picture" idea of what the heck this thing is - and what it can do for us in the PLC programming business ... when I first ran into "y=mx+b" it was presented purely as a mathematical exercise ... there was no explanation of why I needed to learn it - and thus no reason for me to remember it once I'd taken the final exam ... but now I know better ... this thing is a very useful tool ...

so for the "big picture" idea behind how we're going to use "y=mx+b" take a look at the roadmap type "mileage" chart at the top of Figure 3 ...

Figure 3 - Using y=mx+b for scaling - the "big picture" ideas

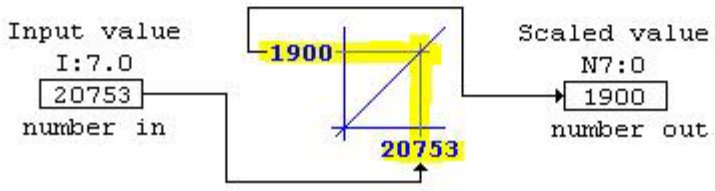
Using y=mx+b for scaling works "sort of - kind of" like this roadmap "mileage" chart ...

Where you want to go	Columbia	Charlotte	Charleston	Fayetteville	Savannah
Columbia	0	88	103	159	158
Charlotte	88	0	189	125	243
Charleston	103	189	0	211	93
Fayetteville	159	125	211	0	262
Savannah	158	243	93	262	0

Where you are now

---

but this is what's really going on ... find your "Input value" at the bottom of the chart ... then go straight up until you hit the line ... then go straight across and read the "Scaled value" on the left side ...



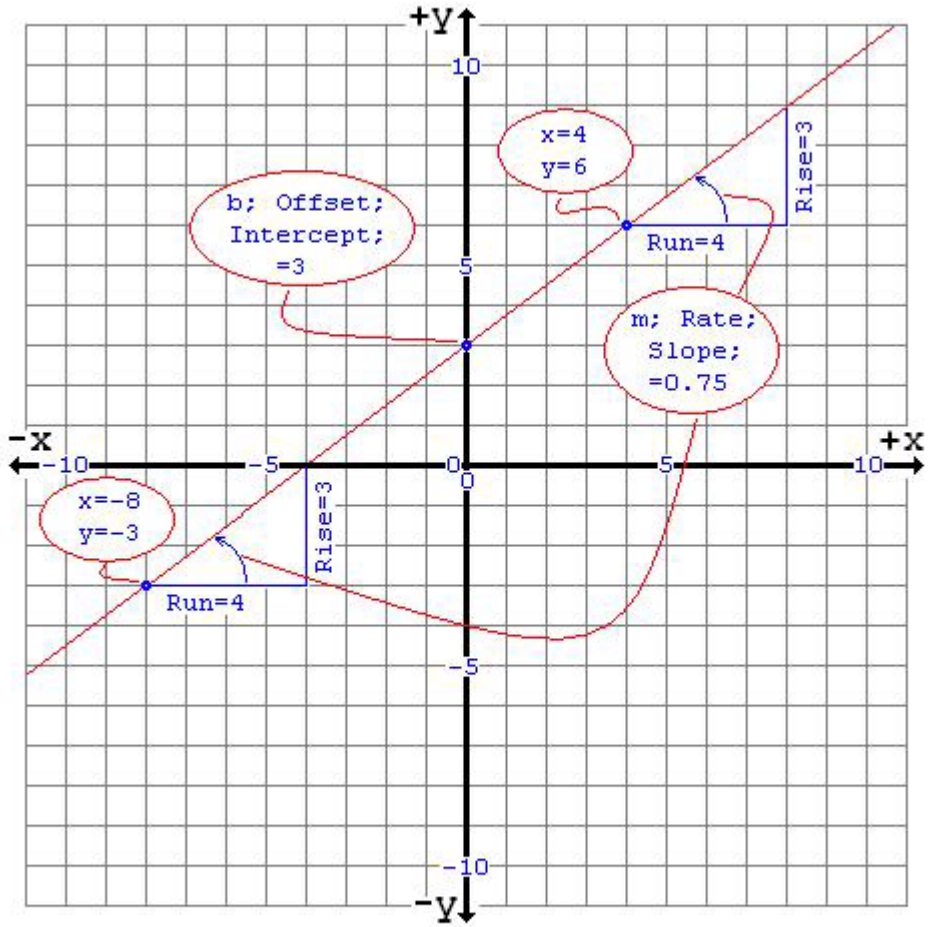
suppose that I'm planning to drive from Charleston to Fayetteville ... I find Charleston in the "where-you-are-now" list at the bottom of the chart ... I find Fayetteville in the "where-you-want-to-go" list at the left side of the chart ... I run one finger along the column and one finger along the row ... I find a number where the two lines cross ... bingo! ...

211 miles ... well “y=mx+b” works “sort of – kind of” like the mileage chart ... but not exactly ...

once you get this thing set up, it’s basically like a straight line drawn on a piece of graph paper ... you find the “incoming” number that you’re interested in somewhere along the bottom (on the x-axis) ... then you go straight up until you hit the line ... then you move straight across until you find a new number at the left side (on the y-axis) ... so the “big picture” is that this thing converts from one number (such as a PLC’s raw input data value) into another number (such as a scaled data value for an operator’s display) ... or said another way, when you shove a “raw” number into a slot at the bottom of the “math box”, then “y=mx+b” spits out a new corresponding “scaled” number on the left side ... this can be quite handy for a lot of different applications ...

now we’re ready to look at the simple “y=mx+b” graph shown in Figure 4 ...

Figure 4 - where the numbers in the formula come from



the formula of the straight line shown here is:

$$y = 0.75x + 3$$

here we've taken a piece of graph paper and marked it up with a horizontal "x-axis" and a vertical "y-axis" ... the best way to think about this type of graph is something like the lines of latitude and longitude on a map of the earth ... we can specify any location on the planet by giving its coordinates of latitude and longitude ... in the same way, we can represent any specific point on this chart by giving its value of "x" and its value of "y" ... usually points like these are specified by giving the "x" value first, then a comma, and then the "y" value ... for example: 4,6 and -8,-3 would adequately define two points shown on our graph ...

suppose that we pick a point located at x=4 and y=6 and mark it with a dot ... suppose that we pick a second point at x=-8 and y=-3 and mark it with another dot ... then once those two points have been marked, suppose that we draw a straight line which passes through both dots and then continues on at each end right across the graph ... as shown at the bottom of the figure, the mathematical formula which defines this particular straight line is "y = 0.75x + 3" ... let's see how that formula works ...

first of all, we can easily see that the line is "sloped" across the graph ... specifically, it's not perfectly horizontal ... and it's not perfectly vertical ... so it's "sloped" at some specific angle ... let's see if we can come up with a number which will perfectly define exactly what angle is involved here ...

first look at the point located at x=4, y=6 ... notice that I've shown a small triangle which starts at this point ... the base (or bottom) of the triangle is 4 squares wide ... now notice that the far side of the triangle is 3 squares high ... this little triangle is all that we need to adequately specify the "slope" of our line ... the "rule" that we'll use to mathematically calculate the "slope" is often expressed as "the RISE over the RUN" ... as shown in the figure, the "rise" is the height of the little triangle ... the "run" is the length of its base ... so in our example, the "slope" of the line can be calculated by saying "3 OVER 4" ... or in other words, "3 divided by 4" ... the answer is, of course, "0.75" ... and that's exactly why the number 0.75 appears in the formula for this particular straight line ...

some people have a hard time with the concept of how just one single number can specify a slope ... but the fact is that the number that we're talking about is a "ratio" (in other words, a "fraction") that takes into account TWO numbers – one number divided by the other ... the same concept is often used by carpenters and roofers when discussing the "pitch" of a roof ... for example, the roof on my new workshop has a pitch of "five-in-eleven" ... this means that for a horizontal "run" of 11 feet, the roof "rises" 5 feet in height ... we could divide 5 by 11 and express the pitch of the roof with just one single number: 0.4545 ...

and common sense tells us that the "slope" of a straight line is always constant throughout its entire length ... that's demonstrated in Figure 4 by the second triangle which starts at point -8,-3 ... this one is exactly the same size and shape of the first triangle ... in fact, you could pick ANY point – ANY where – on the straight line and draw a triangle of exactly the same shape ... the only reason that I chose these two was because they work out to even numbers which are easy to recognize on the graph ... in a little while we'll use my shortcut method on the TI-36X to calculate the "slope" of the line without even having to draw or measure the little triangles at all ...

now one more thing about the little “slope” triangles ... some people wonder “how do you know how big to make the triangle?” ... in other words, how did you know to use a “run” of 4? ... why not some other number? ... well, the truth is you can use ANY number for the “run” and you’ll still get exactly the same “slope” of the line ... the only thing is, it’s a lot easier to see the “slope” when the sides of the triangle work out to even whole numbers ... but try it if you don’t believe it ... for specific examples, start out at our original data point located at -8,-3 ... now go to the right eight squares (for a “run” of 8 ... now count straight up until you hit the line ... you’ll find that the “rise” is 6 squares ... and 6 over 8 (6 divided by 8) equals the same 0.75 “slope” that we’ve already determined for our sample straight line ...

now let’s talk about one of the tricky things that most people find very confusing about this “straight line formula” business ... the fact is that there are at least THREE different names which can all be used to describe EXACTLY the same measurement ... so while we’ve been busy discussing the “slope” of the line, we’ve also been discussing the “Rate” at which it rises ... and in most math books (and in the formula that we’ve been studying) the term “m” is used as a symbol for the “slope” ... why “m” and not “s”? ... no one knows ... maybe the first mathematician who came up with this thing had a girlfriend named “Mary” and he named it after her ... stranger things have happened ... anyway ... the “slope” of the line is EXACTLY the same thing as the value of “m” in the formula “y=mx+b” ... and it’s EXACTLY the same thing as the “Rate” which we ran into for our SCL instruction back in our PLC program ... and that, boys and girls, is why I pressed the SLP (slope) key on my calculator when I wanted to find the “Rate” entry to program into the SCL instruction ... Allen-Bradley calls it the “Rate” ... and Texas Instruments calls it the “Slope” ... and the math formula that makes it all work calls it “m” for some unknown reason ... so how’s that for consistency? ...

well, so far we’ve got the upward angle of our sample straight line pretty well specified by using the value “0.75” for the “Slope” ... or for the “Rate” ... or for “m” ... or for whatever the heck we’re calling it today ... now let’s move on ...

the fact is that if all we happen to know about our straight line is its “Slope” then we’ve got a BIG problem ... specifically, where is the line located? ... in other words, there are a LOT (actually an infinite number) of straight lines which could be drawn on our chart and each one could have exactly the same 0.75 “Slope” as all of the others ... just picture sliding a ruler across the page and drawing one straight line after another by tracing along the edge of the ruler ... just as long as we keep the ruler perfectly parallel to our original line, then EVERY ONE of the lines that we draw will ALL have the same 0.75 “Slope” as the original line ... so what we need next is some way to “nail the line in place” on the graph ... to anchor it so that we’ll know exactly where it is located ... and that’s what the “Intercept” does for us ...

take another look at Figure 4 and find the exact spot where the straight line crosses the y-axis ... we call that particular spot the line’s “Intercept” ... notice that for our sample line, the “Intercept” is located at a point 3 squares up on the y-axis ... and now we’re getting down to business ... because now we know that the “Slope/Rate/m” of the line is 0.75 - and that the line “Intercepts” (or crosses) the y-axis at a

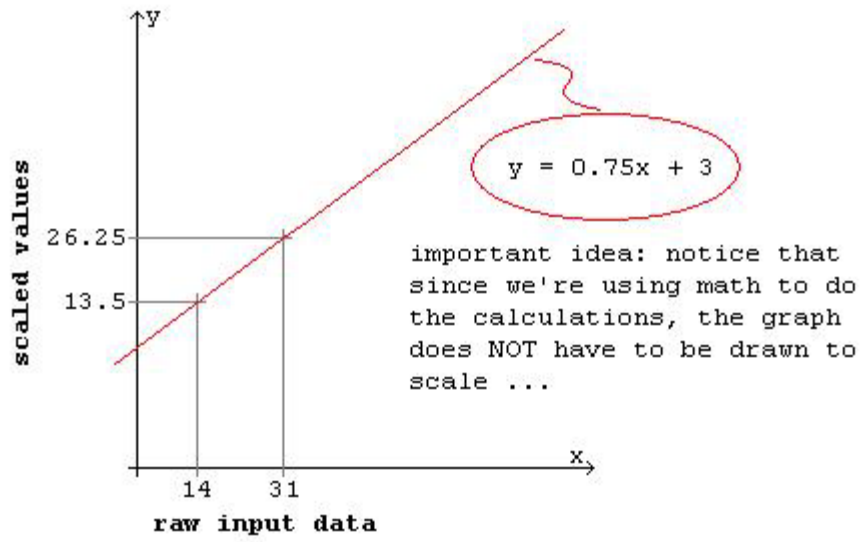
point which is 3 squares up – and so NOW we can absolutely nail down exactly how the line appears on our graph ... specifically, there is one - and ONLY one - straight line which can fulfill the requirements of that one specific “Slope” (0.75) AND that one specific “Intercept” (3) ...

and now let’s talk about another one of the tricky things that most people find confusing about this subject ... unfortunately there are at least THREE different names which can all be used to describe EXACTLY the same “Intercept” measurement ... the “Intercept” can also be called the “Offset” of the line ... and in most math books (and in the formula that we’ve been studying) the term “b” is used as a symbol for the “Intercept” ... why “b” and not “i”? ... yet another mystery from the mathematicians who brought us the formula “y=mx+b” ... it’s no wonder that these people can’t find anybody to talk with at cocktail parties ... anyway ... the “Intercept” of the line is EXACTLY the same thing as the value of “b” in the formula “y=mx+b” ... and it’s EXACTLY the same thing as the “Offset” which we used for our SCL instruction back in our PLC program ... and that, friends and neighbors, is why I pressed the ITC (Intercept) key on my calculator when I wanted to find the “Offset” entry to program into the SCL instruction ... Allen-Bradley calls it the “Offset” ... and Texas Instruments calls it the “Intercept” ... and the math formula that makes it all work calls it “b” ... so, are we having fun yet? ...

now let’s move on and see what our sample formula “y = 0.75x + 3” will do for us ...

Figure 5 shows a quick sketch of the same sample formula “y = 0.75x + 3” that we’ve already studied ... but this little sketch shows only the upper right corner of our previous figure ... this is a very common setup since in most cases (but certainly not in every case) we’re usually dealing only with positive values for “x” and positive values for “y” ... and those “both-positive” values are found only in the upper right corner of the graph ...

Figure 5 - using a quick sketch of the formula



given the formula: y = 0.75x + 3 and an Input value of 31, find the corresponding Scaled value ...

y = 0.75 * 31 + 3 ... substituting into the formula ...

y = 23.25 + 3 ... doing the multiplication step ...

y = 26.25 ... doing the addition step ...

so ... 26.25 is the corresponding Scaled value ...

so here's a simple sketch that graphically represents the "meat and potatoes" parts of the formula "y = 0.75x + 3" ... and one thing that we need to say right from the very beginning is that this particular sketch has NOT been drawn to scale ... in fact, once we have the formula in hand, we really don't need the sketch at all ... the only reason that I'm showing it here is to help you visualize just what's going on in this next little exercise ...

now let's work out a basic "scaling" problem using this little "y = 0.75x + 3" formula ... and we'll work it "longhand" instead of with my "no math" shortcut method ... let's say that a raw data value of 31 comes in on the "x-axis" ... we need to find the corresponding scaled value ... the math steps are listed at the bottom of the figure ... first we substitute the known value of "x" (which is 31) into the equation ... then we multiply the known value of "m" (which is 0.75) times the value of "x" and that gives us 23.25 ... next we do the addition step by adding 23.25 to the known value of "b" (which is 3) ... and so our answer comes out as 26.25

... this is the scaled value which corresponds with a raw data input of 31 ...

now let’s do it the easy “Ron-Beaufort-no-math-shortcut” way ... in fact, let’s go all the way back to Figure 4 and derive the formula “y = 0.75x + 3” from scratch ... and THEN we’ll come back here to Figure 5 and work on it some more ...

before we continue, first take another quick look back at Figure 4 ... this is where we took two data points (point “4,6” and point “-8,-3”) and plotted them on the chart ... then we drew a straight line through those points and graphically derived the formula “y = 0.75x + 3” which we’ve been using as an example ... now let’s do it the easy way by using the TI-36X calculator ... the keystrokes are shown below ... but before we start, we need to discuss a few more new keys ...

some people who aren’t familiar with the calculator run into a problem the first time they try to enter a negative number (for example: “-8”) ... they often hit the “minus” key [-] and then they key in the number ... that won’t work because the “minus” key means “subtract” ... instead you need to use the “change sign” key which is marked [+/-] AFTER you key in the number’s value ... I’ve shown the exact keystrokes in the figure below ...

next notice the key marked [y’] ... this is usually called the “y prime” key but you don’t need to know its name ... just be sure that you press the right one when the time comes ... on the TI-36X this key is also the [+] key used for addition ... but notice that the [y’] legend is shown in yellow ... that means that you need to press the [2nd] key (also yellow) just before you hit the [y’] key ... but then you probably already knew that ... anyway ... I’ve given the exact keystrokes in the figure below ...

what you really need to know about the [y’] key is that it basically means “give me a new number for y” ... it works like this ... we’ll give the calculator a raw data number (on the x-axis) and then hit the [y’] key ... the calculator will give us the corresponding scaled value (from the y-axis) as an answer ...

the same ideas apply to the [x’] key ... it’s usually called “x prime” and it basically means “give me a new number for x” ... seem confusing? ... well you’ll quickly get used to it ... and I guarantee that learning to use these “shortcut” keys is a LOT easier than doing the longhand math ... now let’s tackle the problem shown in Figure 4 again ... Figure 6 below will show us the shortcut keystrokes step-by-step ...

Figure 6 - Using TI-36 Calculator "Stat2" Mode for Scaling

[ON/AC]	turn on and clear calculator
[3rd] [STAT2]	select 2-variable statistics mode
8 [+/-]	value of Input MIN
[X⇆Y]	get ready for next value
3 [+/-]	value of Scaled MIN
[Σ+]	store the MIN values
4	value of Input MAX
[X⇆Y]	get ready for next value
6	value of Scaled MAX
[Σ+]	store the MAX values
[2nd] [SLP]	display "Slope" or "Rate" 0.75
[2nd] [ITC]	display "Intercept" or "Offset" 3
31	raw input data value to convert
[2nd] [y']	display corresponding Scaled value 26.25
13.5	Scaled value to convert
[2nd] [x']	display corresponding Input value 14

pressing [ON/AC] clears out the calculator ... using the [3rd] [STAT2] key combination puts the calculator into the “2-variable statistics” mode ...

the first point has minus values involved but that’s no problem as long as we use the “change sign” key [+/] correctly ... so ...

negative 8 ... [flip/flop] ... negative 3 ... [sum] ...

and we’ve got one data point entered ...

4 ... [flip/flop] ... 6 ... [sum] ...

and the second point is nailed down ... and believe it or not, that's all there is to setting up the problem ...

now suppose that I want to know the "Slope" of the line ... note that I don't NEED to know the slope for the exercises in Figure 5 ... but I'm just trying to impress you with how easy my shortcut method is to work with ... so remember that "Rise-over-Run" division operation that we did earlier? ... my trusty TI-36X has already DONE that calculation for me based on the two data points I've just entered ... watch the magic ...

I press the [2nd] [SLP] key combination to tell the calculator that I want to know the "slope of the line" ... and the calculator instantly displays "0.75" ... NO MATH! ... I just key in the data points and then stroke the keys ... the calculator does all of the math internally ... now THIS is my kind of "statistics" ...

next suppose that I want to know the "Intercept" of the line ... note that I don't NEED to know this either for the exercises in Figure 5 ... but let's see how easy my shortcut method deals with this part ... remember that earlier we graphically found the "Intercept" point by "eyeballing" the spot where the line crossed the y-axis ... that doesn't sound too precise does it? ... well, my trusty TI-36X has already PRECISELY found the "Intercept" point for me based on the two data points I've entered ... all I have to do is ask the calculator for the information ... the magic continues ...

I press the [2nd] [ITC] key combination to tell the calculator that I want to know the "Intercept" of the line ... in other words, where does the line cross the y-axis? ... and the calculator instantly displays "3" ... NO MATH! ... and no "eyeballing" the graph either ... I just key in the data points and stroke the keys ... the calculator does all of the work for me ...

but that's not even the best part ... don't clear the calculator yet ...

now let's go back to Figure 5 and tackle that "scaling" exercise again ... but this time we won't work it out longhand ... instead we'll just finish out the keystrokes shown in Figure 6 ...

so again suppose that a raw data value of 31 comes in ... I key in 31 on my trusty TI-36X ... now I want to know the corresponding scaled value from the y-axis ... so I hit the [2nd] [y'] key combination ... the calculator displays "26.25" ... the exact same number that I found by using the substituting-multiplying-adding longhand steps earlier ... but this time NO MATH! ...

and it even works in reverse too ... specifically, I can convert from a scaled value to a raw data value just as easily as from raw data to scaled ... for example ...

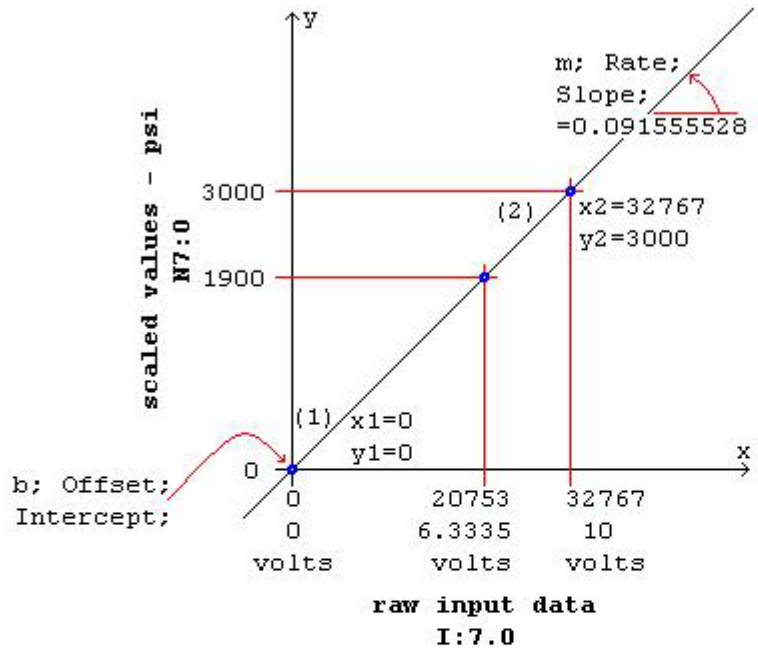
this time suppose that I already know a scaled value on the y-axis ... for example 13.5 ... and suppose that now I need to know what raw data value in the PLC would correspond to that known scaled value ...

I key in the 13.5 known value and then simply press the [2nd] [x'] key combination ... and the calculator instantly displays "14" as an answer ... and again, NO MATH! ...

well, I don't know whether the "Ron-Beaufort-no-math" shortcut approach to scaling has impressed you or not ... but I know that the students in my classes usually get all fired up by this point in the lesson ... in fact, quite a few of them run by Wal-Mart on their lunch break and come back with their very own TI-36X calculators ... that's one reason why I mentioned the TI-30 model at the beginning of this discussion ... when a whole class of my students makes a run on Wal-Mart at one time, sometimes the TI-36X models will sell out ... once in awhile a student will settle for the less-powerful TI-30 model instead – but it won't do 2-variable statistics ... that means there's no magic shortcut for that stripped-down model ...

next we'll work through the simple example shown in Figure 7 ...

Figure 7 – solving y=mx+b by longhand math



$$y = mx + b$$

$$m = \frac{Rise}{Run} = \frac{y2-y1}{x2-x1} = \frac{3000-0}{32767-0} = 0.091555528$$

$$b = y1 - (x1 * m) = 0 - (0 * 0.091555528) = 0$$

and so ...

$y = 0.091555528*x + 0$ is the formula for this line ...

this one is set up for the same conditions used by our original poster ... specifically, we have a 0 to 10 VDC input signal coming in from a transmitter in the field ... that gives us raw data on the x-axis which ranges in value from 0 to 32767 ... we want to convert our raw data values into scaled values which will range from 0 to 3000 psi on the y-axis ...

let's start by making a sketch just to keep the "big picture" ideas straight ... we won't bother keeping this one to scale since we're going to solve everything by math anyway ...

first we draw the x-axis and then the y-axis ... we label the spot where they cross 0,0 ... we'll label the x-axis "raw input data" ... and label the y-axis "scaled values" ... we can go ahead and put a dot to mark a "known" data point at 0,0 ... we can do this because we already know that when the input data (x-axis) is 0, then the corresponding scaled value (y-axis) must also be 0 ... in simple terms, 0 in gives 0 out ... next we'll move to the right on the x-axis and pick a spot (any spot) and label it 32767 ... this will represent a "full-on" reading from our field transmitter ... now we'll draw a line straight up from the 32767 spot for an inch or two and place our second "known" data point ... so make a dot there ... now draw a line straight to the left from that point until we reach the y-axis ... label the left end of the line 3000 ... this will represent a "full-on" scaled value ... specifically, a 3000 psi "scaled" value to go with our 32767 "raw input data" value ... and finally we'll finish up (at least for now) by drawing a straight line through both of the "known" data points ... so basically we've drawn a box with a dot in the lower left corner – and a dot in the upper right corner – and a diagonal line passing through the two dots ...

one of the hardest things about teaching this "y=mx+b" scaling technique to some (most?) students is forcing them to draw this simple sketch ... for some reason they resist doing it ... maybe it's because they don't know EXACTLY how far to go to the right before they pick a spot ... but since we're not trying to keep things to scale, it doesn't really matter how far you go ... any convenient distance will do ... and maybe they don't know EXACTLY how far to go up before they pick a spot ... again, it doesn't really matter ... any convenient distance will work ... all we're trying to do with the sketch is keep the "big picture" ideas straight ...

specifically, a raw data number comes in on the x-axis ... we move straight up until we hit the line ... we move straight over to the y-axis ... and there's the corresponding scaled value ...

so now let's calculate the formula for the straight line shown in Figure 7 ... and this time we won't use the shortcut ... this is pretty much intended for anyone who doesn't have a TI-36X handy ...

first we'll calculate the value of "m" – the "slope of the line" ... as shown in the figure, "m" may be thought of as the "rise" over the "run" ... it may be calculated by "y2-y1 divided by x2-x1" ... pick either of the known points on the graph and mark it (1) ... mark the other known point (2) ... and you can't possibly get this part wrong because it doesn't matter which is (1) and which is (2) ... the formula will work either way ... on my example sketch I've marked the 0,0 point as (1) and the 32767,3000 point as (2) ... so substituting in the formula gives

“3000-0 divided by 32767-0” ... when you work that out, you get 0.091555528 for the “slope” of the line ...

now we’ll calculate the value of “b” ... also known as the “Intercept” ... or the “Offset” ... or the spot where the line crosses the y-axis ... well actually we don’t really need to calculate “b” for this example since we drew the line right through the 0,0 point when we set up the sketch ... but let’s go ahead and make the calculation just for practice ... a popular formula for finding “b” is “y1–(x1*m)” ... substituting in the formula gives “b=0-(0*0.091555528)” ... those “0” terms make it easy ... nothing times anything is still nothing ... and nothing minus nothing is still nothing ... and so “b” works out to be 0 ... well we already knew that ... but just suppose that we had picked the OTHER point to be (1) ... would that have messed things up? ... let’s work it out that way too and see what happens ... substituting in the formula would give “b=3000-(32767*0.091555528)” ... when you work that out you’ll probably get something like 0.000014024 as a value for “b” ... but wait a minute ... that was supposed to come out 0 ... so what went wrong? ... nothing really ... it’s just that the value for “m” that we calculated earlier was “rounded off” slightly ... instead of 0.091555528, the result of 3000 divided by 32767 is closer to 0.091555528427991576891384624774926 ... substitute that monster into our formula for “b” and see if you don’t come up with an answer of 0 ... but then again, why bother? ... 0.000014024 is plenty close enough to 0 for our purposes ... so let’s just say that “b” is equal to 0 and get on with life ...

so now that we know the value of “m” and we know the value of “b”, we can write the formula of the line ... it’s “y=0.091555528*x + 0” ... and of course you can just leave off the “+ 0” part ...

well now that we’ve got that taken care of, let’s use the formula to scale an input value ... suppose that the voltage from the field input device is 6.3335 volts ... that would give a raw data reading of “20753” ... let’s calculate what the scaled value would be ...

since “y=0.091555528*x” is our formula, and “x” in this example is “20753”, then substituting gives us “y=0.091555528*20753” ... and so the scaled value would be “1900.051873” ... and since the result is going to be stored in a PLC’s integer location (N7:0), then we have to round off to “1900” for a final answer ...

and so that’s one way to work out the scaling for a raw data reading of 20753 using “y=mx+b” ... but there are other ways of skinning the same cat ... here’s one very popular method that doesn’t use “y=mx+b” at all ...

take the raw data reading of 20753 and divide it by 32767 ... the “0.633350627” answer tells us that the signal is about 63.3% of “full scale” ... now multiply 0.633350627 by 3000 and you’ll get “1900.051881” ... which rounds off to the same “1900” that we got by using the “y=mx+b” method ...

so which way do you like better? ... at this point, most people want to forget all about “y=mx+b” and go with that last little two-step method ... the only problem is that the simple little divide and then multiply solution will not work for the guy who posted the original question about scaling ... remember that he’s working with an SLC-5/02 processor ... it won’t handle the floating point math required to use the two-step method

we just covered ... specifically, he has no location in his PLC's memory suitable for storing the intermediate value "0.633350627" ... so for his problem, the "y=mx+b" performed by the SCL instruction is the only way to go ...

ok ... as far as Figure 7 is concerned, we did it the hard way ... and we did it another way ... now let's do it the easy way ... I whip out my trusty TI-36X and here are the keystrokes step-by-step ...

[ON/AC] ... [3rd] [STAT2] ...
0 ... [flip/flop] ... 0 ... [sum] ...
32767 ... [flip/flop] ... 3000 ... [sum] ...

and now I've got the two known data points entered ... next I'll scale that raw input data reading ...

20753 ... [2nd] [y'] ...

and the calculator displays "1900.051881" ... this should sound very familiar ... it's exactly the same number that we got when we did the divide-and-multiply two-step solution ... and notice that I got my shortcut answer with NO MATH! ... but that's not all ... suppose that I want to know the "m/Slope/Rate" value ...

[2nd] [SLP] ... and the calculator displays "0.091555528" ... the exact same number as before with NO MATH! ... just multiply that by 10000 and it's ready to plug into the SCL's "Rate" entry ... now suppose that I want to know the "b/Intercept/Offset" value ...

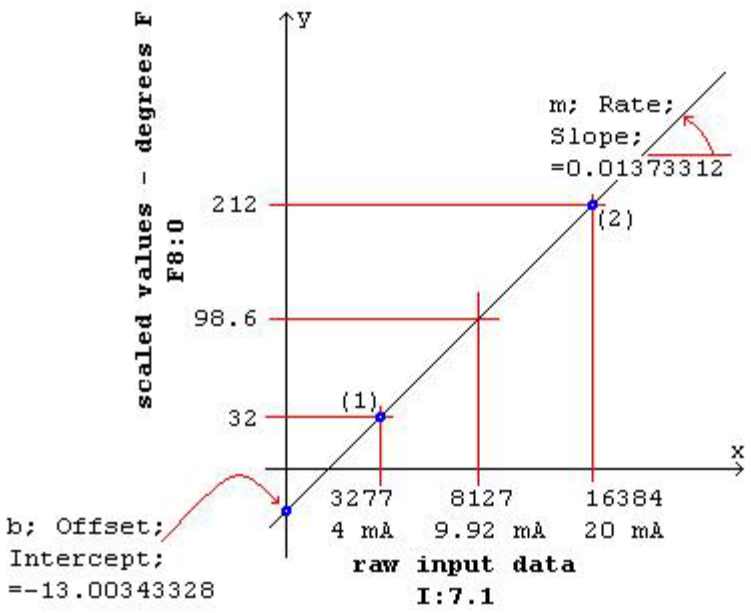
[2nd] [ITC] ... and the calculator displays "0" ... once again, the same solution as before with (you guessed it) NO MATH! ... and it's ready to plug into the SCL's "Offset" entry ...

well, that's enough fun and games with Figure 7 ... now let's tackle something a little bit more challenging ...

the next example uses a different field transmitter setup – one that's probably even more common than the 0 to 10 VDC signal we worked with before ... specifically, this time we have a 4 to 20 mA input signal coming in from a transmitter in the field ... that gives us raw data on the x-axis which ranges in value from 3277 to 16384 ... we want to convert our raw data values into scaled values which will range from 32 to 212 degrees F. on the y-axis ...

once again we'll start by making a quick sketch just to keep the "big picture" ideas straight ...

Figure 8 - an example with an offset on each axis



$y = mx + b$

$$m = \frac{\text{Rise}}{\text{Run}} = \frac{y2-y1}{x2-x1} = \frac{212-32}{16384-3277} = \frac{180}{13107} = 0.01373312$$

$$b = y1 - (x1 * m) = 32 - (3277 * 0.01373312) = -13.00343424$$

note: the apparent slight discrepancy in values for b are due to round-off in the calculator ...

the formula for this line is:

$y = 0.01373312*x - 13.00343328$

first we draw the x-axis and then the y-axis ... we'll label the x-axis "raw input data" ... and label the y-axis "scaled values" ...

we'll move to the right on the x-axis and pick a spot (any spot) and label it 3277 ... this will represent a 4 mA (minimum) reading from our field transmitter ... now we'll draw a line straight up from the 3277 spot and put a dot for our first "known" data point ... now we'll draw a line straight to the left from that point until we reach the y-axis ... we'll label the left end of the line 32 ... this will represent our "minimum" scaled value ... specifically, a 32 degree "scaled" value to go with our 3277 "raw input data" value ...

now we'll move further to the right on the x-axis and pick another spot and label this one 16384 ... this will represent a 20 mA (maximum) reading from our field transmitter ... now we'll draw a line straight up from the 16384 spot and put a dot for our second "known" data point ... now we'll draw a line straight to the left from that point until we reach the y-axis

... we'll label the left end of the line 212 ... this will represent our "maximum" scaled value ... specifically, a 212 degree "scaled" value to go with our 16384 "raw input data" value ...

and that's enough to get us started ... so now let's calculate the formula for the straight line shown in Figure 8 ... and let's do it the hard way ...

first we'll calculate the value of "m" – the "slope of the line" ... as shown in the figure, "m" may be thought of as the "rise" over the "run" ... it may be calculated by "y2-y1 divided by x2-x1" ... substituting in the formula gives "212-32 divided by 16384-3277" ... this one is going to be just a little bit trickier since we don't have those sissy 0 values that just cancel out ... but when you work out the math, you should get 0.01373312 for the "slope" of the line ...

now that we know the value for "m", we're ready to calculate the value of "b" ... also known as the "Intercept" ... or the "Offset" ... or the spot where the line crosses the y-axis ... and this time we can't just eyeball it either ... using the formula "b = y1 – (x1 * m)" and substituting gives us "b = 32 – (3277 * 0.01373312)" ... solve that and you should come up with "-13.00343424" ... and that would be close enough for any scaling that we're likely to do with a PLC ... but ...

note: there is a slight amount of "round off" error in some of the numbers that we're dealing with here ... the basic idea is that most calculators work with twelve digits of resolution – but they can only show ten digits on their display screen ... don't worry about it because it won't significantly affect your final answer ... but just be aware that when the numbers that I show in the figures don't exactly match the numbers on your calculator, it's usually just one of these little "round off" errors causing the mischief ...

so now we can write the formula for our straight line ... "y = 0.01373312*x - 13.00343328" ... and I'm using that particular value for "b" since I know that it's actually closer than the one we calculated just above ... want proof? ... work the math with the Windows calculator ... it has a lot more resolution ... it's just a shame that it doesn't have the built in "magic" for the 2-variable statistics mode ...

well now that we've got the formula written, let's use it to scale an input value ... suppose that the signal from the field input device is 9.92 mA ... that would give a raw data reading of "8127" ... let's calculate what the scaled value would be ...

since "y=0.01373312*x - 13.00343328" is our formula, and "x" in this example is "8127", then substituting gives us "y=0.01373312*8127 – 13.00343328" ... and so the scaled value would be "98.60563296" ... let's just go ahead and call it "98.6" ... and since we're storing it in a floating point location (F8:0), we can keep the decimal point in place ... note that you can't do this in some processors, including an SLC-5/02 ...

and so that's one way to use "y=mx+b" to work out the scaling on this particular system for a raw data reading of 8127 ... now let's try the same simple little "divide-multiply" approach that we used awhile ago for Figure 7 ... fasten your seatbelts - this could be a bumpy ride ...

first of all, we'd like to just divide the raw input data value (8127) by the "full-scale" reading (16384) ... but ... that won't work for this particular problem ... because the input range doesn't start out at 0 for this system ... so before we even touch the raw data value, we need to find the "span" of the input data range ... so we take 16384 (maximum) and subtract 3277 (minimum) and we get 13107 (span) ... now we'd like to just divide the raw input data value (8127) by the span (13107) ... but ... that won't work ... because the raw input data (8127) didn't start out at 0 ... instead it started at 3277 (minimum) ... so we have to take 8127 and subtract 3277 to take care of the "head start" effect ... and we get 4850 ... now we're finally ready to do that first division step ... 4850 divided by 13107 equals 0.370031281 ... this means that our input signal is about 37% of our raw input range ... now we'd like to just multiply our 37% raw signal by the maximum value (212) of our "scaled values" range ... but ... that won't work for this particular problem ... because the scaled range doesn't start out at 0 for this system ... first we need to find the "span" of the scaled range ... so we take 212 (maximum) and subtract 32 (minimum) and we get 180 (span) ... now we're finally ready to do that multiplication step ... 180 (span) times 0.370031281 (signal percent) equals 66.60563058 ... but we're still not finished ... because the scaled range didn't start out at 0 ... so now we need to add in the minimum value of the scaled range ... (we're almost there folks) ... 66.60563058 plus 32 equals 98.60563058 ... so "98.6" is our final answer ...

so what the heck happened to our simple "two-step" divide-and-multiply method? ... it went out the window just as soon as we started working with a system that doesn't use a 0,0 "starting point" for both its input range and its scaled range ... and systems like this are VERY common ...

here's a little history on this particular setup ... once while I was teaching the "y=mx+b" technique to a classroom full of students, there was one crusty old-timer in the group who wanted absolutely no part of the voodoo that I was covering ... he started getting a little bit too rowdy with his criticism, so I politely invited him to show us how easy his favorite "two-step" divide-and-multiply method is when compared to my "y=mx+b" method ... and THIS was the problem that I suggested for an example ... now yes, it CAN be done without "y=mx+b" ... we just did it step-by-step ... but it sure ain't just a quick two-step process of simple division and multiplication ...

so now we've worked through Figure 8 using "y=mx+b" ... and we've worked through it without using "y=mx+b" ... now let's work through it without using any math at all ... specifically, just using keystrokes on my TI-36X ...

[ON/AC] ... [3rd] [STAT2] ...
3277 ... [flip/flop] ... 32 ... [sum] ...
16384 ... [flip/flop] ... 212 ... [sum] ...

and now I've got the two known data points entered ... next I'll scale that raw input data reading ...

8127 ... [2nd] [y'] ...

and the calculator displays "98.60563058" ... bingo! ... and with NO MATH! ...

and just for kicks, I think I'll go the other way ... in other words, let's say that I have a known scaled value of 123 degrees F ... I'd like to know what raw input data value to expect inside my PLC ...

123 ... [2nd] [x'] ... and the calculator gives me "9903.316667" as an answer ... of course that would be "9903" when stored in a memory location such as I:7.1 ... now suppose that I want to know the "m/Slope/Rate" value ...

[2nd] [SLP] ... and the calculator displays "0.01373312" ... with NO MATH! ... now suppose that I want to know the "b/Intercept/Offset" value ...

[2nd] [ITC] ... and the calculator displays "-13.00343328" ... once again, NO MATH! ... and this number is actually a little bit closer to the "real thing" than our previous longhand method because we did one less "rounded off" step ...

and here are the keystrokes I used for the exercises in Figure 8 ... all listed out and ready for you to practice with your very own TI-36X ...

Figure 9 - Using TI-36 Calculator "Stat2" Mode for Scaling

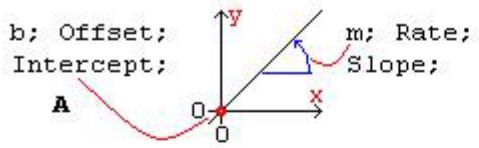
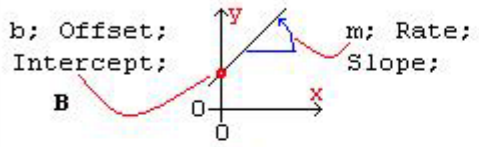
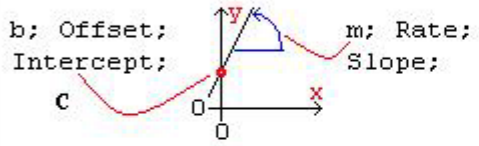
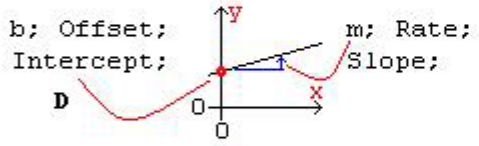
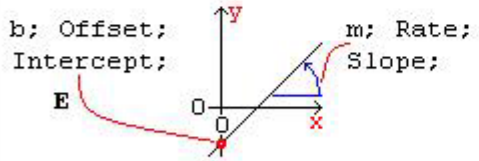
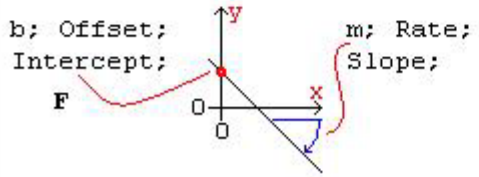
ON/AC	turn on and clear calculator
3rd [STAT2]	select 2-variable statistics mode
3277	value of Input MIN
X⇆Y	get ready for next value
32	value of Scaled MIN
Σ+	store the MIN values
16384	value of Input MAX
X⇆Y	get ready for next value
212	value of Scaled MAX
Σ+	store the MAX values
8127	raw input data value to convert
2nd y'	display corresponding Scaled value 98.60563058
123	Scaled value to convert
2nd x'	display corresponding Input value 9903.316667
2nd [SLP]	display "Slope" or "Rate" 0.01373312
2nd [ITC]	display "Intercept" or "Offset" -13.00343328

well by now you should have realized that there certainly ARE ways to handle scaling problems without using the “y=mx+b” technique ... and in a few cases those methods might be actually easier than using “y=mx+b” ... I’ll freely admit that ... but one of the main things that I personally like about “y=mx+b” is that it ALWAYS works the same – just as long as we’re talking about a straight line relationship between the raw input data and the scaled values that we’re trying to obtain ... this is also called a “linear” relationship in many cases ... and you need to realize that SOME systems out there in the field are not linear (straight line) and so the basic “y=mx+b” method won’t work for them ... but by and large,

the vast majority of analog signal scaling that you'll need to do will be linear and "y=mx+b" will work just fine ...

so here's a chart that gives an overview of just some of the types of scaling problems that "y=mx+b" will definitely handle ... and best of all, once you master the keystroke sequence for my "no-math" shortcut method, all of these different scaling systems can be solved in exactly the same way ... the same pattern of keystrokes works for each and every one of them ... now that's flexibility ...

Figure 10 - if it's a "straight-line" relationship, then with y=mx+b one size really does fit all ...

	here b is located at the origin of the chart (point 0,0); m is the angle shown by the arrow;
	here b is located higher on the y axis; the Slope (m) is the same as in the last chart;
	here we have a higher value for m than we had in the last chart; so if we have the same Input value, we'll get a larger Scaled value;
	here we have a lower value for m; so the Slope or Rate is lower and for the same Input values, we'll get smaller Scaled values;
	this "below 0 Offset" means that some positive Input values (on the x axis) will give us negative Scaled values (on the y axis);
	the negative (downward sloping) line of this chart means that as the Input value (x axis) gets larger, the Scaled value (y axis) will always get smaller;

(discussion continued in part 2) ...

