The Embedded Muse 118

Editor: Jack Ganssle   (jack@ganssle.com)   Oct. 19th, 2005

You may redistribute this newsletter for noncommercial purposes. For commercial use contact info@ganssle.com.

EDITOR: Jack Ganssle, jack@ganssle.com

Editor's Notes

Did you know it IS possible to estimate software schedules accurately? Do you want to increase your team's productivity? Reduce bugs? Meet deadlines? Take my one day Better Firmware Faster seminar. You'll learn how to estimate a schedule accurately, thwart schedule-killing bugs, manage reuse, build predictable real-time code, better ways to deal with uniquely embedded problems like reentrancy, heaps, stacks, and much, much more.

I'm presenting this on two dates:
- Chicago, IL on December 5
- Irvine, CA on December 7

Want to be your company's embedded guru? Join us! There's more info at http://www.ganssle.com/classes.htm, including cheap flights to these cities from around the USA.

If your outfit has a dozen or more engineers who can benefit from this training I can present the seminar on-site. See http://www.ganssle.com/classes.htm .

I've put a new special report up on the web which reviews a number of low-cost USB oscilloscopes and logic analyzers. Unlike traditional stand-alone units, these small (some weigh mere ounces) instruments upload and display the data on your PC. Check it out at http://www.ganssle.com/microscopes.pdf .

The Ganssle Group, www.ganssle.com

More on Naming Convention

David Kellog also attended Jean LaBrosse's Firmware Standards class at the Embedded Systems Conference, and had some thoughts about naming conventions:

The term that I like to use to describe the effective big to small naming of variables is constricting syllables. Each syllable further qualifies those that preceded it. It is important to note that English tends to put adjectives before nouns (i.e., the big bucket). Of course, this heritage works against us in the role of programmers choosing good names. But once recognized, then we can learn how to name things well.

On a related note, in C, `structs` are very under-utilized. Most C compilers carry no overhead for referring to a member of a structure `structs` are just as efficient to access as simple variables. But `structs` provide two important benefits over simple variables:
1. They provide an indication of grouping and association.
2. They automatically provide a sort of constricting syllables way of naming things. In fact, member names can actually be simpler to invent, because they can depend on the parent for some of the description.

For instance, consider `motor.velocity.change` and `motor.velocity.old`. The final member name is not particularly descriptive, but the name as a whole works very well because of the structure of parents. It is obvious that these are related things, because they are part of the same structure.

For a number of years, I too have used the syllable dictionary. Again, this reduces the amount of brain work that I must do to invent a new name. As a matter of preservation, each of our projects is structured with a file named `Abbrev.c`. This is simple a single comment that contains the syllable dictionary. This ensures that future maintenance programmers both understand and can use the naming structure. For example:

```
Abbrev  Meaning
------  -----------------------------
        -- A --
Accl    Acceleration, signed (ref: Dcel)
Acq     Acquiring (ref: L.O.W)
Act     Actual
Ad      A/D (ref: Alog)
Adc     Analog / Digital Converter
Addr    Address
Adj     Adjust
Ai      Analog Input(s)
Aio     Analog I/O
Alog    Analog
Amp     Amps (ref: volt)
```

```
Ana      Analog (ref: Aio, Dig)
Ang      Angle
Ao       Analog Output(s)
Arg      Argument(s)
Auto     Automatic (ref: Manual)
Aux      Auxillary mast, Auxillary pump (ref: Mast)
Avg      Average (ref Typ)

         -- B --
Bad      Bad (ref: Ok)
Batt     Battery  (ref: Bsoc, Pwr)
Beg      Begin (ref: End, Head, Start)
Bld      Build
Bot      Bottom (ref: Top)
Bsoc     Battery State Of Charge
Buf      Buffer (ref: Tbl)
But      Button (ref: Sw)
```

Note the references on some items above, to point to related or opposites. I have found that it is well worth spending several hours crafting a good dictionary that supports the application domain of the embedded system being produced.

Another convention that we have is that acronyms become camel-cased syllables. Only the first letter is capitalized. `Eg,` PWM is used as `PwmOld.` Among other things, this capitalization helps delineate the syllables. Again, it is only an English convention that we write abbreviations in all caps.

Engineering Washouts - Responses

Last issue I referred to an article by Douglas Kern's about being a washout from engineering college (http://www.techcentralstation.com/092105B.html).

People responded! Here are some comments:

Andrew Wyer wrote:
It's a toss up for me. My undergrad is from DeVry Inst. of Tech. By the end of the second trimester we had lost 85% of the students in my class to other programs. Why? Well, more than half of them were only doing engineering because they wanted the money not because they had the skill set. The teachers we had were experienced and pretty focused. I worked with the faculty both as an assistant and a school-employed tutor and got to know them well. About 70% of the time they were doing good jobs, and the same teachers stayed with the same students throughout the program.

I'm entering a masters program at Illinois Inst of Tech and so far I've had mixed experiences. The teachers I've worked with seem to honestly care about students who

show an interest, they are extremely intelligent, and don't obfuscate things intentionally. But until you've established a personal relationship with them they seem to give you the bare minimum and won't give more just because you ask.

Both of my sample points have indicated the same thing: the teachers have about as much interest in your success as the cashier at McDonald's has about your enjoying a hamburger; their goal is just process you through. Once you build a personal teacher-student relationship they change their view to you being the product of their effort.

Its not advice for the schools, but it is advice for other students on how one person got through engineering (by focusing on the liberal-arts ability to make strong interpersonal relationships!)

Rob Fries wrote:
Just a comment on your comments on college/engineering. I'm self-taught (no degree), and have not yet needed advanced math either in the 25+ years that I've been designing embedded systems. Within the past 10 years I've used DSP algorithms and have taken advantage of some of the very useful tools that others have made available for designing FIR and IIR filter coefficients. Some might feel that I've 'cheated' by not taking 4 years of advanced math courses so as to be able to derive these by hand. However, I don't really consider that to be cheating, any more so than using basic trigonometry means that you're cheating because Descartes thought of it first. A good engineer learns from the past and applies the laws of the universe to produce the desired outcome. Aside from DSP, the only math that ever shows up in the work that I do is PID loops. I'm sure in some esoteric branches of the industry there is a need for more math, and I applaud the folks who can do that, and I wish I had more time to get myself up to speed in those areas. Mainly because of what my high school math teacher, Mr. Bloxam, said: "Math is one of the few things that would exist even if we did not."

Barry Lord wrote:
It expresses exactly what I found on a course "Natural Sciences" at Cambridge back in the 70s. After excellent teaching at school, the abysmal equivalent at university (with few exceptions) came as a shock. The options allowed a broad-based course, so I took Chemistry, Physics and Biology of Cells, specializing later in Physics. I finished the course, probably because I didn't have enough imagination then to give up and do something more useful.

Thirty years later, I do work in engineering, but I struggle to remember a single useful thing I learned from three years at university. I should have gone out and got a job.

Prof. Bruce Jacob wrote:
I couldn't agree with you more. I do not believe that the solution is to teach teachers how to teach (that's a nice mouthful, isn't it?)... When I taught high school math, I saw that people who majored in a particular topic in college tended to do a better job teaching it than people who had education majors (learned how to teach). I think the same is true for good designers -- really good designers aren't good because they were taught well, they just are. (Spontaneous generation?)

What you can do to improve the level of design across an organization (as opposed to within an individual) is to remove bad habits, encourage good habits, catalog "known-good" solutions and encourage/ mandate heir reuse, etc. It requires the right carrot -- i.e., management has to demand design quality, recognize it when they see it, and reward the designers appropriately. I think the same idea can be applied to engineering education ... dangle the right carrot, and you'll get the right results, and the effort is more easily applied at the organizational level than the individual level. You said it best: "the current practice of hiring and promoting teachers based on their research and PhD credentials is fatally flawed." Research universities do not consider themselves to be centers of education; they are businesses, profit centers -- the carrot encourages us to bring in as many research dollars as possible (even if it means doing marginal research) and write as many papers as possible (even if nobody reads or cites them). Right carrot if your goal is to self-perpetuate, wrong carrot if you want to educate engineers.

To improve engineering education requires a completely different model ... I'd argue that you would have to restructure a college/ department from the ground up. Rather than filling the ranks with people who are good at getting dollars, you have to fill the ranks with (at least) three different classes of people who demonstrate strengths in complementary areas: research (paper-writing and money-getting, which could very well be two different types of people), development (people good at building things), and teaching. Very few people are good at more than one of these, and I know of only a handful good at all three. The current university structure is designed to retain only the first, which weakens the program in two ways: without good teachers, education suffers. You've already pointed that out. But also -- without people who are good at building things, a college/department's research program is by definition only theoretical -- which is decidedly not what engineering is all about. Engineering (at least, in my view) is wonderful particularly because it represents a healthy blend of theory and practice -- the abstract and the concrete. A good engineering-education program would recognize quality in all three of these areas and hire/retain anyone displaying any one of them. I know of no program in this country that does this, and I have more than half a mind to try to start one. :) What I don't know is how much demand there would be, and whether the target audience would be able both to discern and to appreciate the difference between such an organization and a top-25 engineering university ... Would a prospective future engineer be willing to forsake a degree from Berkeley, Stanford, MIT, Illinois, Michigan,

etc. to go to a no-name engineering school that (nonetheless) does it better than anyone else?

Clifford Wignell wrote:
You asked what do you think, so here are my thoughts.

I am the product of the military training system (Australian Army) basically the training provided was most of what is covered in the first of the Bachelor of Engineering well at least down under. The course is heavenly focused on practical engineering, little calculus and a lot of practice. I have a knack for maths (I know it's a birth defect) and find the more esoteric maths interesting but I have never had to use any thing more complex than Trig professionally!

The largest number engineering I have managed is 28, these ranged from university graduates to crusty old timers. All I can say it the degree programs are missing courses on Work Ethics, Acting like an adult and common sense. I have also been surprised by the difference in apparently identical degrees, one engineer I employed had covered real time control and machine vision and other from a different university was barely capable of using a CRO, both very bright but very different skills.

The company I work for hinted that my potential was limited because I did not have a degree, so a few years ago I set about to get one, I was more than horrified to sit for 40 minutes a listen to someone who could recite the material but did not understand it; had no love for it and could not teach. I have seen a fellow inmate at one of these session fall out of a chair, asleep. In the Army I was taught about getting and maintaining the attention of my students, I was also taught that it is the height of rudeness to teach when you don't fully understand yourself. I believe if you truly understand a subject you can take a person at random off the street and teach them, no boot camps to filter people who may require a little work.

A great deal of my time, a considerable sum of money later, I have a degree; I don't feel any better equipped to do what I do and I don't feel any smarter!

I don't know how to fix the degree system, some in this country are very good, for engineers RMIT is one of be best, it boils down to the lecturers. Good ones are priceless.

Jim Scandale wrote:
There are two major reasons why this happens, neither of which are at work in Douglas Kern's case. (a) bad teachers - I am a teacher so I am very aware of the ones who don't even like the job and therefore do it badly. (b) terrible preparation - especially in math in high school. In fact it is so bad that I can honestly say that in many cases, it is no

preparation at all.

Mr. Kern's case is a bit different - he was in exactly the wrong curriculum, a fact to which he gives a slighting acknowledgement when he says "My accomplishments all pointed towards a more verbal course of study" than Chem-E. We all have delusions about our abilities, some more than others. He ran against his delusion as an undergraduate and executed a course correction.

Joerg Schulze-Clewing wrote:
What the author experienced is kind of normal. At our university we had professors who bragged about how many students they were able to fail this time around. University isn't so much for learning; it is there to make you tough. Just like the army. Ok, this sounds harsh, but in business only the tough ones make it.

Tenure is a problem. There is a saying that tenure breeds incompetence. I don't know, I believe teachers should follow their calling. A teacher does not need tons of credentials, we are already harping too much on credentials in this country. He or she needs to enjoy teaching, and then it'll all come together.

Insisting on credentials is one of the reasons why we are short on school teachers. Why don't they let engineers in? Most of my best teachers had no teaching credentials. I never had formal management training yet my boss entrusted me with running a division of about 100 people, with full P&L responsibility. Don't want to brag here but I guess I did ok, considering that the company turned around and our production problems were largely history about a year and a half into that job. Years later a European corporation liked what they saw and paid top Dollar to buy our company. Wish I could have continued, oh well. It was fun. You need to love what you do, not some license or other paper that says you are entitled to do it.

Last but not least the author might not have understood one important detail: Assuming that the current trends continue America does not need more engineers. So what's his worry?

Dave Kellogg wrote:
You are dead on target about teachers that can't teach, and managers that can't manage. [There is also a 3rd truism that I won't go into here: parents that can't parent.]

In my experience, less than 10% (perhaps closer to 2%) of managers have any true management skill. They may know a lot of technical stuff, but most are clueless about the people skills needed to effectively leverage the people reporting to them. The sad fact is that most managers are chosen (ie, promoted) for being good at technical stuff, not

because of their potential for successfully managing. And then even worse, they are not trained in the management skills after being promoted. Small wonder that Dilbert is so popular.

A truly good teacher has a natural gift for teaching. Yes, teaching can be a learned skill, but only within the bounds of the raw meat that a person is born with. Just as with managers, there are precious few good teachers perhaps 1 in 20. Research has practically nothing to do with being an effective teacher.

Both management and teaching can be learned (to some extent) by apprenticeship and intensive critique and feedback. Both disciplines would benefit from learning from the feedback of those below them (students or employees). Unfortunately, 360 degree review is not widely appreciated for its true value. Extensive role playing in front of a video camera with subsequent playback and honest review can be very effective.

Michael Covington, Ph.D wrote:
Very good point about the traditionally poor quality of teaching the mathematical sciences, especially engineering. This is a field in which you'll keep having students, and some of them will even succeed, no matter how badly you teach. The job market motivates them, and some of them can learn the material from the textbook alone. The rest, you can dismiss as slackers.

I'm *really* against "weed-out courses." The place for weeding out is the admissions process. A university shouldn't take people's time and money if it doesn't expect them to succeed.

I'm also against needless competition. Learning is not a competitive sport. Students should have incentives to help each other succeed, not beat each other in an artificial rat race. In my courses, I make it clear that I expect everyone to succeed, and that no one benefits if someone else fails.

Finally, I think another sign of a sick curriculum is "courses that are their own prerequisite" (my daughter's apt term) -- that is, courses that people normally take more than once in order to pass. It's all too easy to say "we have weak students" (then why did we admit them?). We should be saying, "we need better teaching."

Michael L. Anderson wrote:
I have to defend the use of advanced math and physics in the engineering curriculum. This is the stable foundation upon which all other work will be built. Specific technologies (software languages, hardware families, etc.) come and go much too rapidly. The student must get a real taste of what is hot at the moment, but s/he must also get the

basic foundation. It may sound stuffy, but an EE who has no idea what Maxwell's equations are is not a true professional.

I will likewise defend the need for a basic 'liberal' education for anyone who is considered to be educated.

Yes, academia is remarkably poor at striking this balance. I see no easy fix for this. I am HIGHLY worried about where I am going to hire my next technical people.

Sharon M. Foster wrote:
In response to Douglas Kerns's article I can only say, Amen, brother!

I just wish I had done what he did and changed course. I, too, was good in math and science in high school. I came from a family of engineers, and even though my standardized test scores were tilted toward verbal skills, and my personal proclivities pointed to a life surrounded by art and books, the emphasis at home was on making a "good living", and engineering was the way to do that, at least so it seemed in the 60s.

I struggled through college physics, chemistry (only one year of it, thankfully), and math. I usually managed to "get" the math about a semester after I "had" the course, no thanks to the teacher. I squeaked by with a 2.9 average. After getting all As and Bs in high school, it was a rude awakening.

When I graduated I fell into embedded systems quite by luck, and now, 25 years later, I find the work drying up at a frightening pace. I am finally bailing out and going back to school to get an M.S. in Library Science. I probably should have done it 10 years ago, right after I reached the zenith of my career when I worked for the company that made the mirrors for the Chandra X-Ray Telescope, but better late than never, as they say.

Stephen F. wrote:
I too read the Douglas Kern piece and found it typical of engineering education. Engineers work really hard and take lots of hard classes, but in the end not much to show for it.

I looked at my class load and found that I was a class short for a math minor and a physics minor. And the administration is surprised that it takes 5 years for engineers to get through the program. The university was trying to get more students through faster and put a credit limit on the colleges. Then engineering schools ended up doing tricky things like making Math 122 required (which required Math 121 Calculus but not mentioning that in the syllabus.)

Most of the time I found that I was having to learn the material on my own and with study groups. Too often the Professor and TAs were not very helpful, though they got better the higher I got. Grading on a curve might make sense in things like social science and English Literature, but is insane in the math and sciences. The whole school system needs to be rebuilt from the ground up.

The sad part is that we learn a lot of math, a lot of physics, and all the things about our particular branch of engineering and what do we get for that, a windowless cube, a modest salary, no real chance of advancement, and for the love of Pete don't let the engineers near the customers.

Sure, engineers are intelligent, but I wonder if we are smart.

David Telling wrote:
When I got out of the USCG in 1973, I had the GI Bill to help pay for college, and it was my intent to start an engineering course. However, because of other personal considerations, I decided to start with an Engineering Technology course from Capitol Radio Engineering Institute (CREI). At this point, I had been fooling with electronics for about 11 years, I had a Technician ham license (hated CW!) and had spent the last three+ years as an electronics tech in the CG. However, the first course dealt with math and magnetics, and I immediately fell off of a cliff. I had done poorly in algebra in school, OK in geometry, but for some reason I wasn't getting it. The magnetics was not much easier. I still don't totally understand what all of the different terms actually mean (I can plug numbers into a formula pretty well, though!) and I have subsequently found out that I really didn't need them much.

After a couple of lessons, it became clear that I was not doing all that well, so I figured that I was just stupid, and that I should stay with the technician gig. I did this for several years, until I went for an interview with the Fairchild Hybrid Products Division. They wanted a senior tech, and after the interview, I got a call from them saying that they wanted to hire me as an assistant design engineer. Score! I worked for them for about 2 ½ years, got good reviews, did some cool design work (I designed & built a wire-wrapped 12-bit arbitrary function generator that used an HP85 as the controller, for example) BUT was absolutely prevented from a promotion because I didn't have a degree. By this time, the 10 year window for the GI bill was almost out, and since I knew that I was a math idiot, I figured it would not be worthwhile to even try.

The next thing I know, one of the FCI product engineers is telling me that Mallory (the aftermarket ignition company) is looking for a design engineer. I decided to give it a try, and in October of 1982, started the job that I still have (albeit with a new name - Mr. Gasket). What I found was that I was able to do the design work because I had a good understanding of electronic components & circuits. I also found that I liked programming

microcontrollers (this had been kind of side hobby for many years) and started incorporating them into Mallory products. Several years into this job, for some reason I decided to take college trig. I had never taken a trig course, and thought that since I did OK in geometry, trig would not be too bad. In fact, I found that I understood the concepts, but the algebra really made the work tough. I slogged through it, and found that I got pretty good at it. Subsequently, a few years after that, I decided to tackle calculus, as that seemed to be the math that separated the engineers from the rest of the world. I did take a prep algebra/trig course & got A's in that, and to my surprise got pretty good grades in calculus. What was interesting is that I found out why I had such trouble as a kid - no adult ever made any effort to find out why I was having so much trouble. My teacher would write, "David could do better if he would just try harder" and I was infuriated because I just did not "get it", especially converting story problems to formulae. I had a good teacher here at the local community college who helped me get through some of the tough spots, to the point that one of the other teachers asked if I had considered going for a math degree!

At any rate, I started looking into getting a degree, but ran into another stumbling block - I tried very hard to get some kind of work/experience credit at our local university (UNR) but got nowhere. I could challenge some of the non-engineering courses, but nothing else. When I discussed this with the assistant Dean of the engineering school, he agreed that it wasn't fair, but schools had those rules to maintain their income. In addition, I heard enough stories similar to Douglas Kern's about TA who couldn't speak English, professors who were arbitrary in their grading, etc., that I just decided to forget the whole thing. The only real regret I have is the number of people with whom I come into contact who think that if you don't have a degree, you are worthless for ANY technical position, regardless of your experience. At Fairchild, I was doing work that would normally have been done by a graduate engineer, and in fact supervised techs with associate degrees. I had a good boss who based his assignments on demonstrated ability, and I am forever grateful to him for that. Here at Mr. Gasket, I get to do a bit of everything, and as long as the company is still in business, I plan to be designing electronic products.

If It's Not On Fire

Sharon Foster contributed this interesting and amusing story:

I was working at Hughes-Danbury Optical Systems as a temp, while going to school at night. I was responsible for the software for the Precision Metrology station, which used an interferometer to measure the surface of the mirrors that Hughes was fabricating for the Chandra X-Ray Telescope (then known as AXAF), and it was the best job I'd ever had. Not only was it close to home, but it was the nearest I'd ever gotten to a space project. Often I would have to leave for class in the middle of investigating a problem,

and I would happily come back to the lab after school to work until midnight or later. Everyone on the station had my home phone number, and I welcomed them to call me at any time if they had a problem that they needed help with. I was not in on the program at its inception, so there was a lot about the equipment that I didn't know, but I did know the software, and I knew what it controlled and what it did not control. Frequently I was able to either pinpoint a software problem, or rule out a software problem, which would send them off on the correct path to a diagnosis.

With six mirrors being made, the fabrication and metrology stations ran around the clock, and a log book was kept at each station in order to facilitate communication among all the engineers and technicians on the job. Each morning on my way in, before I went to my desk, I stopped at the PM station to read the log entries from the night before. At this point we were having daily station meetings, and I liked to be prepared, both to explain the software, and to defend it against false accusations from people who didn't understand the various components of the station as well as I did.

One morning I came in to read a notation that there had been a problem with the Fluke data recorder. "That's odd," I thought, "no one called me about this last night." The Fluke recorded 40-channels of various analog and digital data, and most failures were initially assumed, except by me, to be a software problem. In fact, one day the technicians had taken the Fluke apart for some routine maintenance, and when they put it back together, it didn't work anymore. After some hours of troubleshooting, it was revealed that the I/O boards had been put back in the wrong slots. Hmph! Some software problem! So I couldn't wait to hear what had gone wrong this time.

The station software read the Fluke data through a serial interface, but the Fluke also produced a hardcopy on thermal paper tape. The paper had jammed, heated up, and the Fluke literally caught fire! That must be why no one thought it was a software problem as there were visible flames and smoke! If it's on fire, it's not a software problem. Ergo, if it's not on fire, it must be a software problem. At last, I had an important insight into the workings of the [non-software] engineering mind!

Postscript: When I left Hughes-Danbury, the PM station crew gave me a fire extinguisher as a going-away present. Affixed to it was a plaque with Fosters Theorem: If it's not on fire, it's a software problem.

Jobs!

Let me know if you're hiring firmware or embedded designers. I'll continue to run notices for embedded developers as long as the job situation stays in the dumper. No recruiters please.

AMCC/3Ware is a leading provider of high capacity, switched Serial ATA (SATA) RAID storage solutions. Our groundbreaking SATA hardware RAID controllers enable a completely new class of RAID storage solutions for high performance computing applications. If you want to join our dynamic and world-class firmware engineering team delivering leading edge storage products, please send your resume to ctherene@amcc.com.

We have the following opportunities:
- Staff FW Engineer - Hardware-related activities
- Principal FW Engineer - Performance-related features
- Staff FW Engineer - Sustaining activities
- Firmware Engineering Manager - RAID Firmware Group

Programming in C and C++ experience is required. In-depth knowledge and experience with the development of storage products is required - SAS/SCSI, ATA/SATA and/or RAID.

Please visit us at http://www.amcc.com/Employment/jp_Sunnyvale,_CA.html

Embedded Windows CE Developers (ref # 4002g) Vancouver, BC
Want to take the next steps on the leading edge of technology? You will be responsible for requirements analysis, system design, and new state-of-the-art implementations of MS Windows CE & Mobile (CE, Pocket PC & Smartphone) board support packages (BSPs). Perform system design and implementation of new custom BSPs. Participate on development teams for 3 to 6 month projects. Lead small development teams. Provide technical opinions, advice; flag and resolve issues and risks; and provide mentoring to junior Project members. Work on customer sites as required (some international travel may be required)

Ideally you have 3 or more years experience including: Requirements analysis, design and development of Window CE and/or Linux BSPs, driver development experience for Window CE and/or Linux, and strong working knowledge of C++. If you don't have direct low level experience in Windows CE & Mobile we will train you.

www.intrinsyc.com for info and to apply, quote the job reference number and job title in the subject heading.

Joke for the Week
Catherine French contributed this one:

Why Engineers and Scientists Don't Write Cook Books

Chocolate Chip Cookies

Ingredients:

1) 532.35 cm3 gluten
2) 4.9 cm3 NaHCO3
3) 4.9 cm3 refined halite
4) 236.6 cm3 partially hydrogenated tallow triglyceride
5) 177.45 cm3 crystalline C12H22O11
6) 177.45 cm3 unrefined C12H22O11
7) 4.9 cm3 methyl ether of protocatechuic aldehyde
8) Two calcium carbonate encapsulated avian albumin coated protein
9) 473.2 cm3 theobroma cacao
10) 236.6 cm3 de-encapsulated legume meats (sieve size #10)

To a 2-L jacketed round reactor vessel (reactor #1) with an overall heat transfer coefficient of about 100 BTU/F-ft2-hr, add ingredients one, two and three with constant agitation. In a second 2-L reactor vessel with a radical flow impeller operating at 100 rpm, add ingredients four, five, six and seven until the mixture is homogenous.

To reactor #2, add ingredient eight, followed by three equal volumes of the homogenous mixture in reactor #1. Additionally, add ingredients nine and ten slowly, with constant agitation. Care must be taken at this point in the reaction to control any temperature rise that may be the result of an exothermic reaction.

Using a screw extrude attached to a #4 nodulizer, place the mixture piece-meal on a 316SS sheet (300 X 600 mm). Heat in a 460K oven for a period of time that is in agreement with Frank & Johnson's first order rate expression (see JACOS, 21, 55), or until golden brown. Once the reaction is complete, place the sheet on a 25C heat-transfer table, allowing the product to come to equilibrium.

About The Embedded Muse

The Embedded Muse is an occasional newsletter sent via email by Jack Ganssle. Send complaints, comments, and contributions to him at jack@ganssle.com.

To subscribe, send a message to majordomo@ganssle.com, with the words "subscribe embedded *your-email-address*" in the body. To unsubscribe, change the message to "unsubscribe embedded *your-email-address*".

The Embedded Muse is supported by The Ganssle Group, whose mission is to help embedded folks get better products to market faster. We offer seminars at your site offering hard-hitting ideas - and action - you can take now to ***improve firmware quality and decrease development time***. Contact us at info@ganssle.com for more information.