Build Your Own

EFIS, Part 1

An introduction to building and programming an Electronic Flight Instrument System.

BY JAMES P. HAUSER

Over the course of the next month or so, a series of three articles about a build-it-yourself Electronic Flight Instrument System (EFIS) will be presented; the articles will include both hardware and software techniques. Although intended for the neophyte, even those readers expert in avionics systems may find these articles to be of interest; they are condensed from my book "Building and Programming Electronic Flight Instrument Systems." The articles following this one will appear on www.kitplanes.com.

The primary focus of these articles will be on the use of open source software and open architecture hardware. This means the software source code is available for potential modification by the builder and the hardware is IBM PC-compatible. The former provides flexibility and ease of upgrading; the latter provides lower-cost hardware.

From time to time, I will post additional information on my company web site—www.aerospectra.com. Although not in any way connected to my company, there is also a web site devoted to this concept at www.openefis.org that may be of interest.

A natural question might be, "Why would I want to build my own EFIS when so many fine systems are available ready-made with some even certified?" The answer is of course the same as for building your own aircraft. It's educational, you'll gain the satisfaction of building your own equipment, and it can be customized to your preferences. There may even be financial savings, although this would likely be negligible if the builder's time were to be considered. But "build-your-own" considerations aside, these articles will also provide a better understanding of how these systems work.

One very important caveat: Some inspectors will not certify an amateur- built aircraft that does not have approved primary instruments. This generally means the instruments are manufactured under FAA Parts Manufacturing Approval (PMA) to the appropriate Technical Standard Order (TSO). In this case, a homebuilt EFIS could only be used as a VFR advisory system. From a safety viewpoint, this is a very wise and prudent position on the part of the inspector.

In this first article, an overview of a typical EFIS will be presented. It will also illustrate the minimum that you may expect from a build-your-own EFIS.

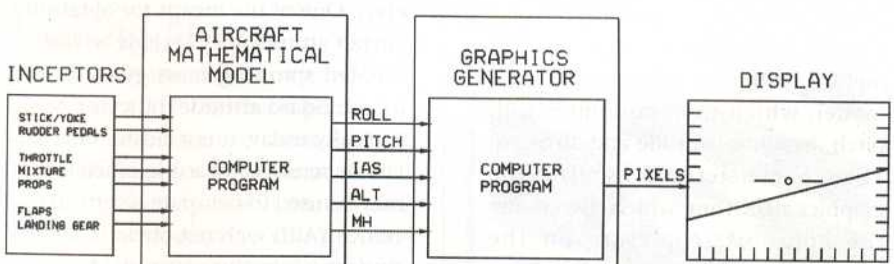

Figure 1.

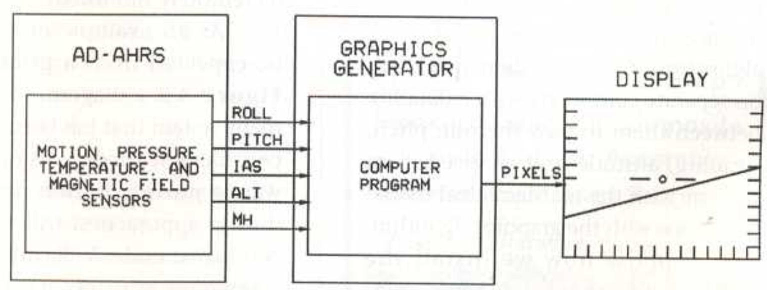

Figure 2.

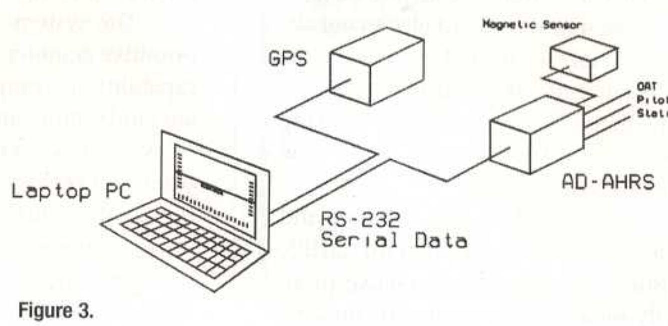

Figure 3.

ILLUSTRATIONS: JAMES P. HAUSER

Build an EFIS
CONTINUED

PC Flight Simulator and EFIS Similarities

Most readers will have already had some experience with a PC-based flight simulator; some of the current offerings include very detailed terrain rendering. In appearance, these simulations differ little from the high-end synthetic vision EFIS systems available from some avionics manufacturers. With the necessary information, some of these simulators may be tailored to perform identically to real aircraft.

Figure 1 is a block diagram of a typical PC-based flight simulator. For simplicity, navigation and terrain rendering functions are not shown.

The inceptors (control stick/yoke, throttle, rudder pedals) provide information to the aircraft mathematical model, which then computes roll, pitch, heading, altitude and airspeed. These parameters are input to the graphics algorithm, which then draws the appropriate picture on the computer terminal.

Suppose the flight simulation program consisted of two separate programs: one, the mathematical model of the aircraft, and the other the graphics algorithm. We could then run them on separate computers with a datalink between them to pass the roll, pitch, heading, altitude and airspeed from the one with the mathematical model to the one with the graphics algorithm.

Suppose now we install the graphics algorithm on a laptop computer. Furthermore, figure that we have sensors/transducers installed in our aircraft to provide the roll, pitch, heading, altitude and airspeed data required by the graphics algorithm. After making the appropriate connections, we now have a primitive EFIS.

Enter the Air Data, Attitude and Heading Reference System (AD-AHRS, usually pronounced *ad-a-hars*); properly installed in the aircraft, the AD-AHRS is the heart of any EFIS. The AD-AHRS takes the place of the mathematical model and its computer that simulates the motion of the aircraft; it also produces measurements of the actual motion of the aircraft in which it is installed.

Figure 2 is the block diagram of an EFIS using an AD-AHRS and a graphics algorithm similar to a flight simulator program.

So, what is inside an AD-AHRS? Generally less than one might imagine. The air data is provided by an absolute pressure transducer for static port pressure, a differential pressure transducer for pitot pressure and an outside air temperature probe for OAT. Indicated airspeed, rate of climb, true airspeed, pressure altitude, barometric altitude and density altitude all may be computed from these fundamental parameters. One of the means for obtaining aircraft attitude may include an instrumented spinning mass gyro, similar to a standard attitude indicator. More typically today, it is a cluster of angular rate sensors and accelerometers that may be used to compute a vertical reference. (Although not obvious, a laser-ring gyro is in this category.) Magnetic heading is obtained from a magnetic field sensor, which may or may not be remotely mounted.

As an example of what might be expected from a primitive EFIS, **Figure 3** is a diagram of a development system that has been flown with considerable success. Its primary goal was to provide runway visualization during approaches. (All flights were conducted under VFR conditions with a separate pilot/observer operating the laptop computer.)

The system currently employs primitive graphics to demonstrate the capability to compute wind and the aircraft landing point. Other map features, such as roads, significant structures and provisional targets can be included to further augment situational awareness during flight in low-visibility conditions. If a Traffic Collision Avoidance System (TCAS) is installed in the aircraft, this information could also be displayed in a 3D format. Video, IR and radar are other optional information sources that could be combined with the computer-generated view.

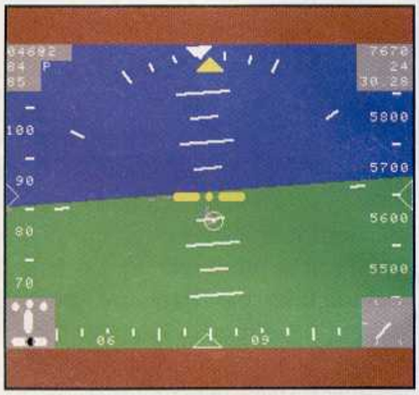

Figure 4.

Operational Example

As an operational example of the capabilities of this system, **Figure 4** is a screen image captured during the beginning of an actual final approach to Runway 08 at Boulder, Colorado. The resolution of this image is relatively low (320x240), as it was intended for a small instrument panel display; larger displays would enable much higher resolutions and a crisper appearance.

Wind direction/speed, groundspeed (knots) and TAS (knots) are displayed in the upper-left corner. Density altitude, OAT and altimeter

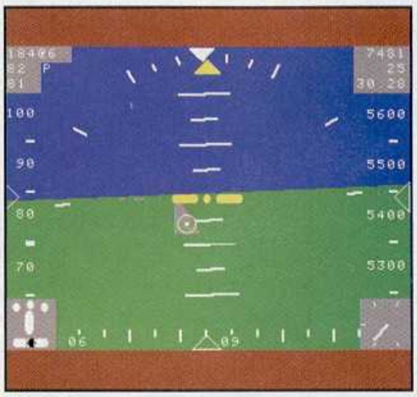

Figure 5.

setting are in the upper right. In the lower left is a conventional turn and slip indicator; in the lower right is rate of climb. The left scroll indicates indicated airspeed (mph), and the right scroll indicates altitude. The lower horizontal scroll is magnetic heading. The target symbol (white circle with dot) is the computed touchdown point. The green bar across the approach end of the runway is a glideslope indicator presented in a form similar to VASI lights. Two other airports may be seen in the distance.

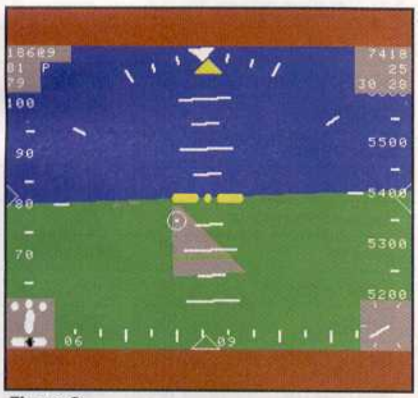

Figure 6.

Figure 5 is the screen image on final approach at about 150 feet AGL. Notice that the runway is now larger in relation to the screen. Note also the drift correction angle of about 5°; heading is 085, but the track is 080.

Figure 6 depicts the beginning of the transition to the flare at 100 feet. Rate of descent is slowing, and the target moves up on the screen during the transition. At this point, the pilot would be flying by visual reference to the ground.

While the above images are primitive in appearance, they illustrate a simplified synthetic vision EFIS.

Adding Terrain

As an example of terrain depiction, **Figure 7** shows a simple graphic rendering of the terrain southwest of Boulder using USGS DEM data. For illustration purposes, the vertical relief is exaggerated. Again, the resolution

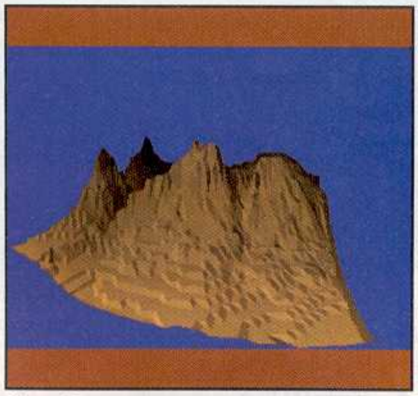

Figure 7.

was kept deliberately low. Near photo-realistic rendering at high resolution is possible at low cost with today's computer technology. Including terrain information would also enable the inclusion of a Terrain Avoidance Warning System (TAWS) into the EFIS.

Looking Ahead

In this installment, a very basic overview of EFIS has been presented. The next installment will introduce the reader to the software necessary for an EFIS. The approach will be directed toward the neophyte, so no previous programming experience will be required or expected. The third installment will review the hardware necessary to build an EFIS.

To read Parts 2 and 3, visit www.kitplanes.com. Part 2 will be posted on about June 1; Part 3 will appear around June 15. ✈

Dr. James P. Hauser is a graduate of the USAF Academy and former military pilot. He is a registered Professional Engineer in the state of Colorado and is the president of AeroSpectra Inc., a consultant engineering firm.

Build Your Own EFIS