

体感型プラネタリウムを製作して夜空を探検しよう



システムの記事



ARMの記事

第1回 ARM基板と有機ELタッチパネルを使った星空ナビゲータの製作

山崎尊永

本稿は、本誌2008年5月号付属ARM (Cortex-M3) 基板や有機ELタッチパネル、GPSモジュールなどを用いて、夜空の星の位置が分かる星空ナビゲータ「StarFinder」を製作し、それに利用した技術を紹介する。本機器は、ARM基板と市販の電子部品、フリーのソフトウェアのみで開発しているので、個人で製作できる。プログラムはすべて本誌のWebサイト (<http://www.cqpub.co.jp/dwm/>) からダウンロードできる。短期集中連載の第1回は、星空ナビゲータの概要と製作方法について解説する。次回以降で、天体位置計算の原理やハードウェア制御方式、ソフトウェア構造などを詳しく紹介していく予定である。

(編集部)



写真1 星空ナビゲータ「StarFinder」の外観

本機器は、本体を向けた方向に見える天体をディスプレイに表示して星空を案内してくれる、夏の星空探訪には欠かせないグッズである。透明アクリル・ケースに、タッチパネル付き有機ELディスプレイ、6軸センサ(加速度・地磁気)、GPSモジュール、時計IC、マイコン、電池などが入り個人で製作できる。天体位置計算には多くの浮動小数点演算を実行するが、本機器に使用したマイコンCortex-M3 (72MHz) は比較的高い性能を発揮してくれた。

これからの夏の夜空は、銀河系の中心方向(いて座)が見えます。空気のきれいな場所に旅行したときなどはきれいな天の川が期待できます。あるいは2008年も8月12日～13日に期待されるペルセウス座流星群の放射点を見極めたいかもしれません。

そういうときのため、筆者は、簡単にどの方角のどの高さに、どんな星や星座が見えるのかを案内してくれる星空ナビゲータ「StarFinder」を製作しました(写真1)。本機器を見たい方角と高度の星空に向ければ、その場所に見える各種天体をカラー表示の星図の上にリアルタイムに表示します。星座早見盤よりも直観的に星空の情報を得られる、まさに体感型の天然プラネタリウムです。

連載第1回の今回は、StarFinderの使い方と製作方法を詳しく説明します。本機器は本誌2008年5月号付属ARM基板や有機ELタッチパネル、GPSモジュールなどで構成されています。プログラムのソース・コードは本誌のWebサイト (<http://www.cqpub.co.jp/dwm/>) ですべて公開します。今後、天体位置計算の方法、内部のソフトウェアの構造などを順次説明していく予定です。今回の記事だけでも十分に星空散歩を楽しめますが、さらに機能拡張した、コンテンツなども追加予定です。

ちょっとりコストがかかる部品もありますが、その分、製作やプログラム開発で大いに楽しめます。「大人の電子工作」として取り組んでみてはいかがでしょうか。そして、ぜひ、宇宙の大きさを感じながら、お子さまやご家族、恋人と一緒に星空の散歩を楽しんでください。

Keyword

有機EL, OLED, タッチパネル, 4D Systems, GPS, NMEA, FV-M8, 6軸センサ, 3軸加速度センサ, 3軸地磁気センサ, TDS01V, RTC-8564NB, Cortex-M3, STM32, 星空, 天球, 天体位置計算, 恒星, 惑星, 地平座標, 赤道座標, 緯度, 経度

1. 「StarFinder」の概要

表1に「StarFinder」の仕様と天体位置計算方法の概要を、図1に天球座標と本機器の向き関係を示します。また写真2には使用している様子を示します。星空観測時は、本機器は左右になるべく傾けないように向きを合わせてく

ださい。

今回使用するのは、本誌2008年5月号に付属したSTMicroelectronics社のARM Cortex-M3コア搭載マイコン「STM32F103」です。内蔵ROMが128Kバイトしかないので、プログラムを詰め込むのに苦労しましたが、演算が高速で(72MHz)かつシリアル系のインターフェースが多いので、デバイスとしては使い勝手が良かったと筆者は

表1 「StarFinder」の仕様

天体位置計算の詳細やソフトウェア構造の詳細などは、本連載の中で順次解説していく予定。ソフトウェア仕様(コンテンツ)については、今後も追加予定である。

分類	項目	内容
ハードウェア	有機EL表示デバイス	4D systems社「uOLED-32028-PMD3T」: 抵抗膜型タッチパネル付き有機ELディスプレイ(OLED), 2.83インチ, QVGA(320×240), 26.2万色
	6軸センサ	バイテック「TDS01V」: 3軸加速度+3軸地磁気センサ(ほかに、気圧センサ、温度センサ、電圧センサ内蔵)
	GPSモジュール	Sun Jose Technology社「FV-M8」: 32チャンネル, 5Hz更新, バッテリ・バックアップ
	RTC (リアルタイム・クロック)	セイコーエプソン「RTC-8564NB」: バッテリ・バックアップ
	RS-232C インターフェース	Dサブ9ピン・オス・コネクタ: OLEDプログラム用, GPSモジュールのデータ転送用(個人的には、天体望遠鏡に接続して自動導入に使用)
	使用マイコン	STMicroelectronics社 ARM Cortex-M3「STM32F103」
	電源スイッチ	プッシュONタイプ: シャット・ダウンは、画面メニューから行う
	電池	メイン用: 単4アルカリ電池×4本, または単4ニッケル水素電池(2次電池)×4本 バックアップ用: CR2032×1個 そのほかの機能: オート・パワーOFF機能(10分間)
	消費電流	メイン・バッテリー: 動作時200mA, 停止時1μA未満 バックアップ電流: 1μA未満(電容量1000mAhの充電式ニッケル水素電池を使うと、連続で約4時間動作可能)
ソフトウェア	6軸センサ・データ表示	生データ/平均データ/最小値・最大値表示: 地磁気センサの校正データのフラッシュ・メモリへの保存
	電子コンパス	方位角/高度/傾斜表示: 方位角の計算のためには、3次元地磁気ベクトルの偏角と伏角が必要である。これらは観測地点によって変化する。日本付近に関しては、緯度・経度の値から、近似式で偏角・伏角を求めており、ほぼ正確である。日本付近を離れた観測位置においては、偏角はゼロ、伏角は緯度の値を基に概算で求めている。
	時計表示・設定	地方標準時・協定世界時(UTC)・時差設定
	観測地設定	観測地の、緯度・経度・高度の設定: フラッシュ・メモリへの観測位置の保存
	GPS機能	GPS測位データ表示と、時計と観測地設定への同期: フラッシュ・メモリへの観測位置の保存
	Sky Walk 星空案内	本機器が向いている方向のリアルタイム星図表示機能: 正距方位図法で表した地平座標系の中に、恒星(名称)・惑星・太陽・月(月齢)・星座名を表示。
天体位置計算方法概要	時刻の扱い	時刻は協定世界時UTC(Coordinated Universal Time)が土台。ここから地球の自転の遅れを補正した一定の割合で推移する力学時TD(Dynamical Time)を算出して天体位置計算を実行。内部的には、ユリウス日や、グリニッジ視恒星時などを計算している。本機では、恒星時系の世界時UT1については、UTCとの差が0.9秒以下なのでUTCと同一とみなしている。
	内蔵する恒星データ	恒星のデータは、ヒッパルコス星表(Hipparcos Catalogue)をもとに独自に作成した。4.5等星までの894個の恒星データ(赤経・赤緯・等級・名称・バイエル名)を、マイコンのROMに内蔵してある。恒星は赤道座標上で常に固定の場所にあるわけではなく、時とともに少しずつ移動している。これは、恒星自身の固有運動や、地球の自転軸の変化(歳差、章動)、年周視差、光行差などによる。ほとんどの項目を計算で求めることができるが、固有運動に関するデータがROMを圧迫する。ここでは、2008.08分点における各恒星の赤道座標をROMに格納している。指定した日時における恒星赤道座標の計算方法は本連載で今後紹介予定。
	太陽位置計算	太陽は、力学時TDの関数として、ほかの惑星が地球に及ぼす摂動力を考慮した黄道座標系の黄経を算出して視位置を求めた。超高精度計算は不要なので黄緯はゼロとみなした。光の伝搬遅延時間による光行差も考慮している。
	惑星位置計算	太陽系の惑星はその軌道が摂動などにより変化するため、ここでは力学時TDの関数として得られる変動する平均起動要素の値を使って、2体問題として位置計算している(ケプラーの方程式)。水星・金星・地球・火星・木星・土星・天王星・海王星の太陽系内での3次元位置を計算している。地球の位置を計算するのは、ほかの惑星との視関係を求めるため、光行差も考慮している。
	月の位置、月齢の計算	月の位置は、力学時TDの関数として、地球やほかの惑星が及ぼす摂動力を考慮した黄道座標系の位置を計算して、視位置を求めた。月は近いので、地心から見た視位置ではなく、観測点の測心位置から見た視位置を計算している。光行差も考慮している。月齢は、太陽と月の黄経差から、収束法で求めた。
	大気差	地平線近くでは、光の屈折によって天体が浮き上がって見える。これを大気差という。本機器では天体の真高度から大気差を計算して、視高度の位置に天体を表示している。
	浮動小数点演算	内部演算はdouble(64ビット)で実施。ROM内のconst定数は容量節約のためfloat(32ビット)で格納。