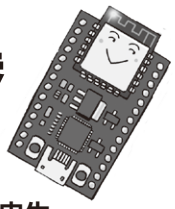
ダウンロード・データあります

700円マイコンESP32ではじめる 逆引きMicroPythonプログラム集



第6回 定番SPI/I²C/UARTによるシリアル通信

角 史生

表1 ESP32のハードウェアSPIバスのピン配置
ハードウェアSPIは最大80MHzと高速動作に対応するが，ピン配置の変更はできない．ソフトウェアSPIは動作が最大40MHzに制限されるが，任意のGPIO出力ピンに割り当てられる

信号名	ピン番号	
	HSPI (ID=1)	VSPI (ID=2)
SCK	14	18
MOSI	13	23
MISO	12	19
CS	15	5

表2 SPIクロックとMODEの関係
SPIは制御対象ごとにクロックのpolarity（極性）とphase（位相）が異なるので，初期化時にはMODEの設定が必要になる

項 目		phase（データ取り込みタイミング）	
		0（SCK奇数番目のエッジ）	1（SCK偶数番目のエッジ）
polarity（IDLE状態のSCK）	0（SCK：“L”）	MODE0	MODE1
	1（SCK：“H”）	MODE2	MODE3

MicroPythonは，リソースの少ないマイコン上でPython 3と同じようにプログラミングできる環境の実現を目指して開発された言語処理系で，プロトタイプ開発に向いています．

プロトタイプ開発では，試作，テスト，修正を繰り返しながら開発を進めますが，MicroPythonを用いることでトライ＆エラーが容易になります．本連載ではESP32-WROOM-32（Espressif Systems）を搭載する開発ボードESP32-DevKitC（Espressif Systems，以降はESP32と表記）を使って，用途別にMicroPythonの使用例を紹介します．

● 定番のSPI/I²C/UARTはMicroPythonでも使える

ESP32では，定番のシリアル・インターフェースであるI²C，SPI，UARTの3種類が使えます．本稿では，これらの使い方を解説します．プログラムの具体的な実装例は，各項目のソースコードを参照してください．文献(1)にも周辺I/Oの実装例が紹介されているので，興味のある人は参照してみてください．

5-1 4本の通信線で数十Mbps! SPI通信によるデータ送受信

● ESP32のMicroPythonでは2種類のSPIが使える

SPI（Serial Peripheral Interface）は，通常4本の信号線で構成されるシリアル・インターフェースの一種です[注1]．制御対象との通信速度は数Mbpsです．

ESP32のMicroPythonには，ハードウェアSPIとソフトウェアSPIの2種類が用意されています．ハードウェアSPIには，HSPIとVSPIの2系統が用意されています．ハードウェアSPIは，最高80MHzで動作しますが，**表1**に示すピン配置から変更はできません．

ソフトウェアSPIは，最高通信速度が40MHzに制限されますが，ピン配置は変更可能です．出力可能なGPIO端子であれば，どこにでも割り当てられます．

■ ハードウェア（接続方法）

SPIでは，制御対象のデバイスが複数台ある場合でも，同一バス上に接続できます．

複数のデバイスを接続している場合，通信したい対象デバイスのSS（Slave Select）信号またはCS（Chip Select）信号をESP32のGPIOを使って有効（Enable）に設定します．通信したい対象デバイスが1台の場合は，対象デバイスのSSを“L”に固定して，常に有効にすることも可能です．

本連載では，SDカードの制御をHSPIで行い，グラフィックス・ディスプレイの制御をVSPIで行った事例を紹介しています．SDカード制御は次回，グラ

注1：一般的なSPIバスは，MOSI，MISO，SS，SCKの4線式ですが，マスタ-スレーブ間通信をMOSIで兼用させる3線式もあります．