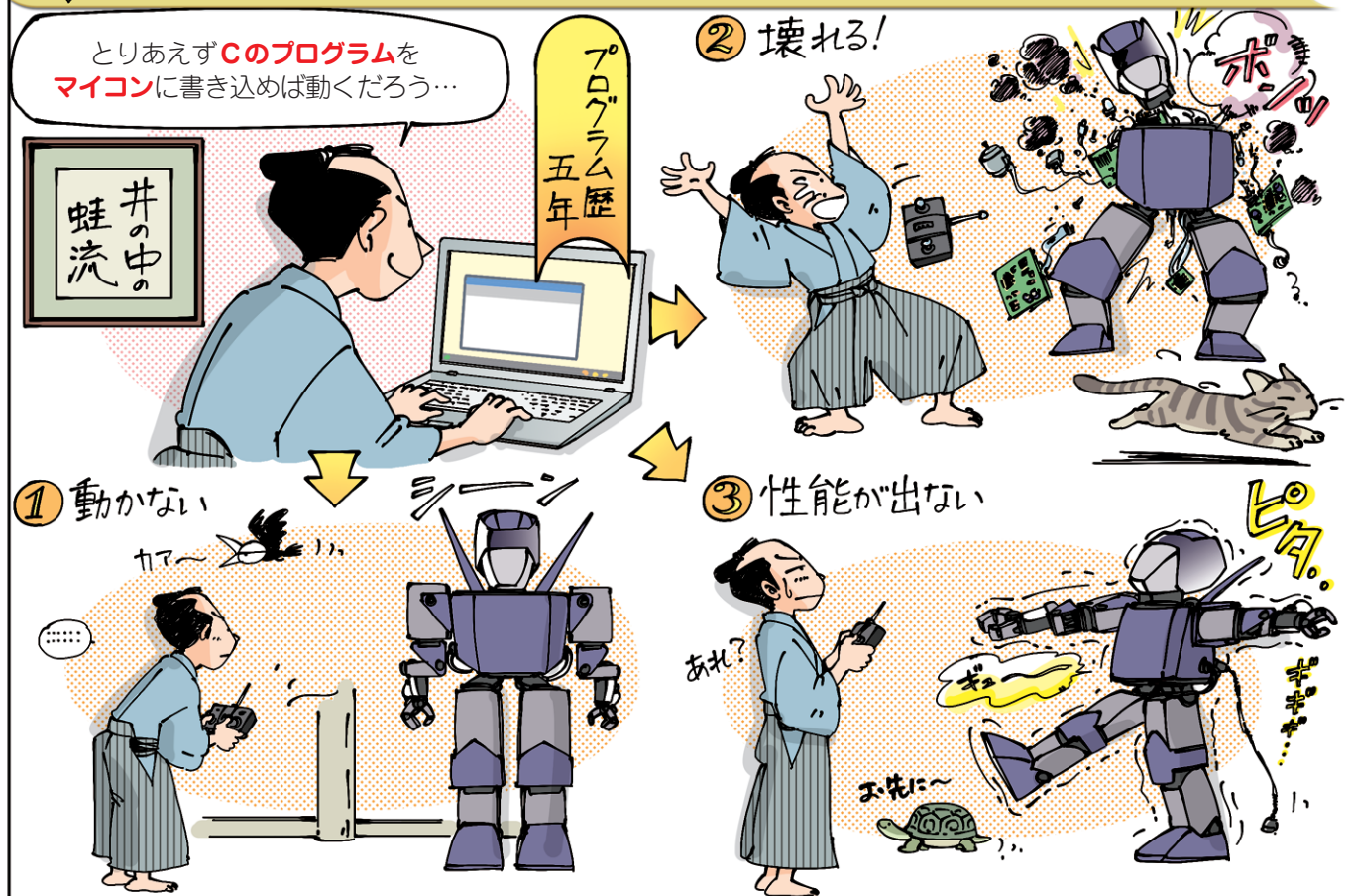




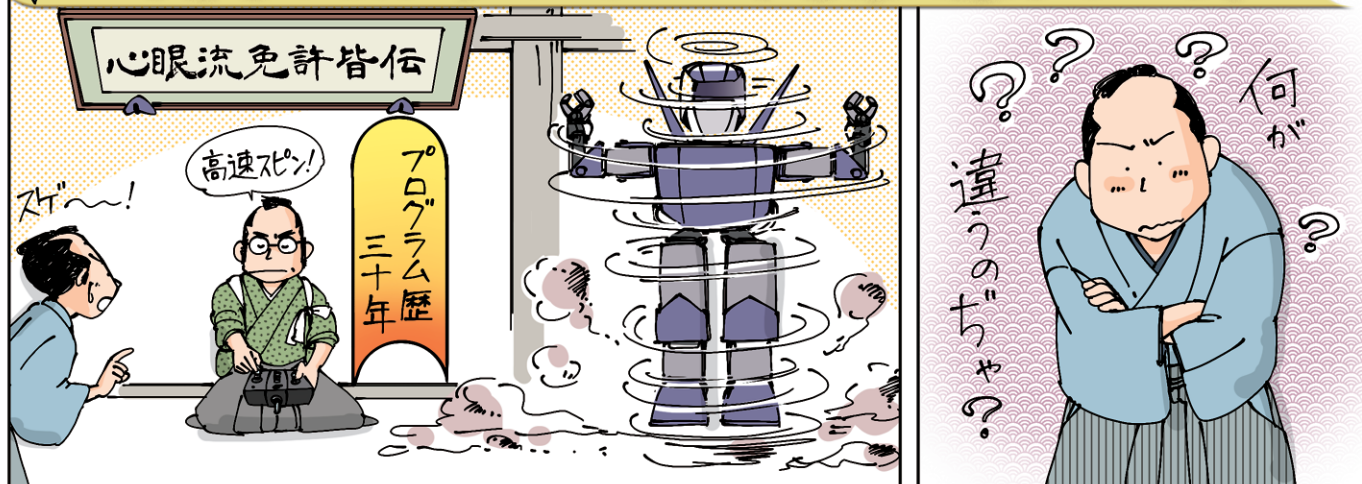
ハードの動きを感じて一発で



ハードを知らないプログラマがはまる落とし穴



できるプログラマは、ハードを常に感じている!



決める プログラミング

編集部



Intro

App1

App2

1

App3

App4

2

3

4

App5

App6

App7

App8

5

App9

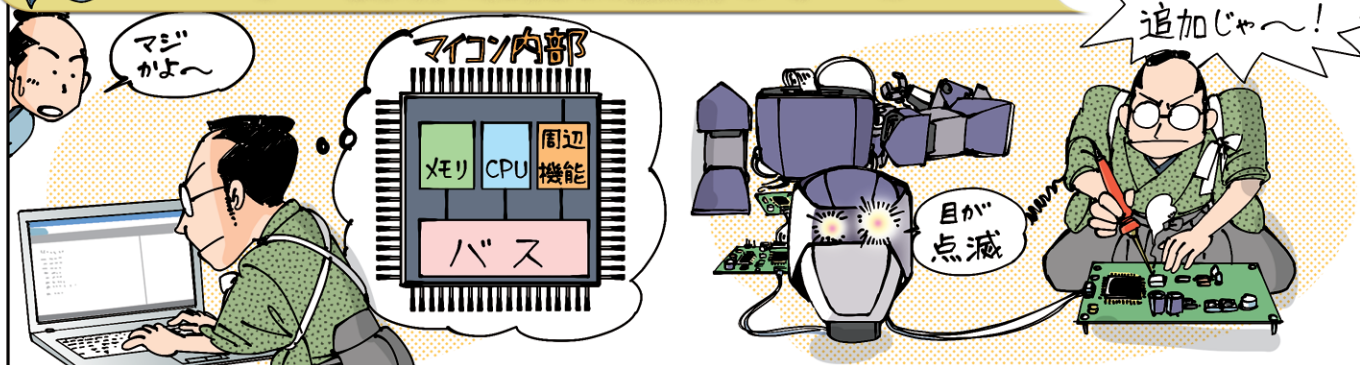
6

7

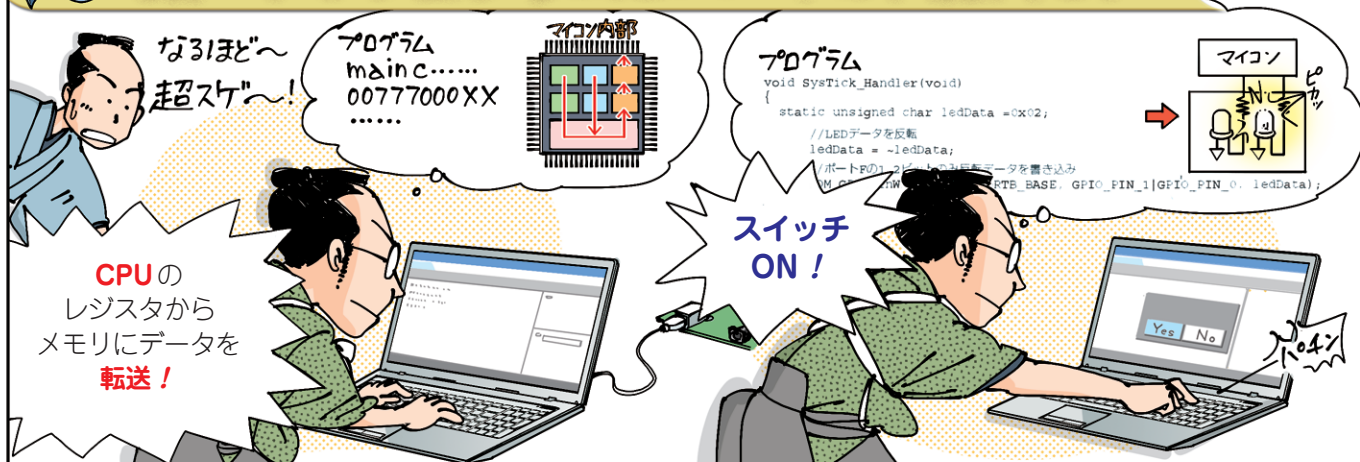
8



ハードの動作環境を理解している



頭の中でプログラムとハードが同期する!

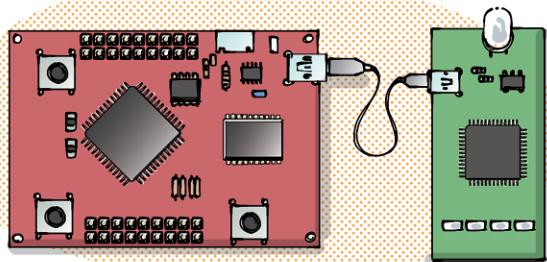


特集では！



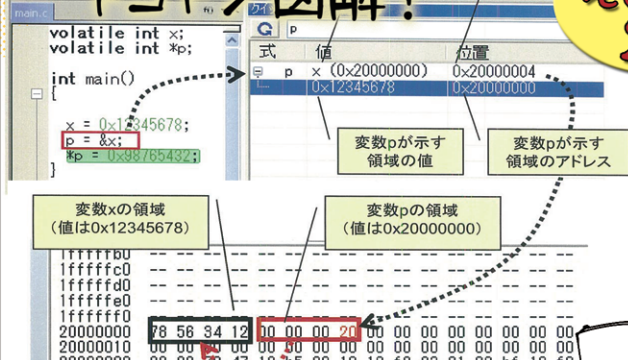
✕ トコトン実験で
ハードの動作を解説

見どころ
1



✧マイコン内部の動作を
トントン図解!

見どころ
2



✧ プログラムとハードの動きを同期させて解説!

見どころ
3

```
//ポートBへのクロック供給
//この設定がないとポート・レジスタにアクセスできません(例外処理が発生する)
ROM SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
```

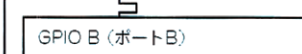
```
// 方向レジスタの設定(1=出力)  
// 第1引数=アドレス 第2引数=出力にしたい端子  
// デジタル端子イネーブル・レジスタの設定もしています  
ROM_GPIOPinTypeGPIOOutput(GPIO_PORTB_BASE, GPIO_PIN_1|GPIO_PIN_0);
```

```
// データ・レジスタの1ビット目のみを1に設定(LEDはHアクティブ)  
// 第1引数=アドレス 第2引数=操作したいビット(複数の場合はOR)、第3引数=書き込む値  
// 関数内ではリードライトモディファイしています  
ROM_GPIO_Write(GPIO_PORTB_BASE, GPIO_PIN_1|GPIO_PIN_0, GPIO
```

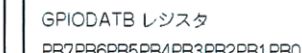
⑦ポートBにクロック信号が供給される



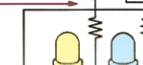
②ポートの初期設定



インピーダンスからポート



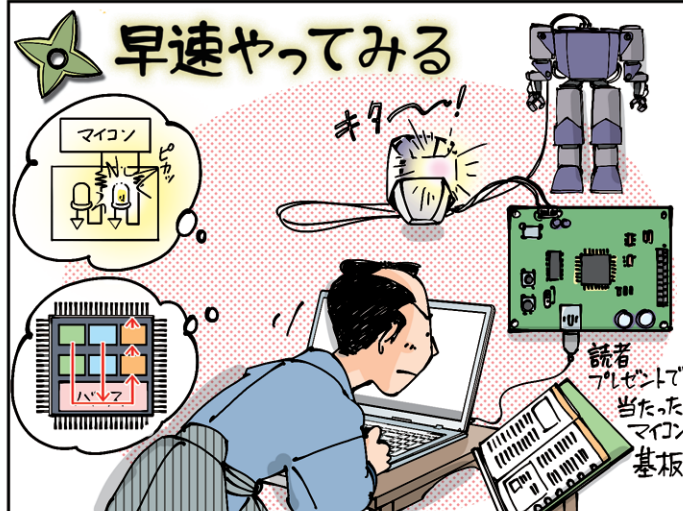
④ポート・レジスタのPB0に '0' を、PB1に '1' を書き込む



同期



★早速やってみる



もうハードは恐くない! 

