



## 無須撰寫程式碼

# 輕鬆將觸控與近距感測功能整合至嵌入式系統

By Mark Lee, Principal Applications Engineer, Cypress Semiconductor Corp.

### 介紹

電容感測操作介面在近年來發展成一項實際且創新的技術，在許多種類的消費性產品中取代按鈕開關，例如在平滑表面的發光區域，或是偵測手指靠近的元件，都能取代機械按鈕開關。

電容感測模組讓業者輕易在系統中加入觸控或近接感測介面，因為它不需要撰寫程式碼也不必進行除錯，僅須把現成的模組加入系統即可。感測器模組所執行的功能都是固定的，缺乏調整彈性，可能因此限制了產品擴充功能的空間。業者需要彈性較高的解決方案，因為常常需要在最後一刻在產品中加入新功能，以提高競爭優勢。

本文介紹如何把電容感測功能整合到一個簡單的嵌入式系統，過程中連一行程式碼都不必撰寫。免撰寫程式碼的模式，發揮現成模組的便利性，以及結合含有大量程式碼嵌入式控制器的彈性。完整的設計流程，涵蓋實際產品從設計理念，一直到開發出可運作原型產品等階段。

免撰寫程式碼的系統設計模式，採用Cypress提供的PSoC Express軟體以及FirstTouch評估套件。設計新產品的創意人員，不必再浪費無數心力來開發可運作的原型產品。若您能畫出系統的模塊圖表，就能利用PSoC Express的圖形化介面開發出實際產品。即使是PSoC Express的新手，也能在數小時內完成整個設計流程，可降低開發時間與專案成本。

### 打造更好的捕鼠器

最經典的發明目標就是打造更好的捕鼠器。下面討論的設計例子很接近最後的結果。我們要設計一個嵌入式系統，在設計例子中結合電子產業的兩項趨勢：電容感測與數位照相。我們用的例子是一個智慧型近接感測器，用來啟動一個數位攝影機。這種系統稱為“相機陷阱”。在近幾年來，這些攝影機一直是科學新聞的頭條。許多野生動物研究人士利用相機陷阱來證明許多一度被認為絕種的動物仍在野外生存，像是亞歷桑那州的美洲豹。在像是波羅州濃密森林這類荒郊野外，科學家還利用這類攝影機發現許多新物種。現有的攝影機陷阱技術似乎相當完善。我們何必要發明新的科技？以下扼要地比較新舊技術之間的差異，並闡述新技術如何打造出更好的陷阱。

#### 舊技術：

採用PIR動態感測器。在四週溫度較高的環境，在目標區域內的動物會消失在背景環境中。反射的陽光會產生偵測錯誤的狀況。笨重的硬體設備，須妥善隱藏以躲過動物的目光，並防範各種破壞因素，視線中出現任何物體，就會讓觸發裝置失效。使用者很難透過更換鏡頭的方式來調整觸發區域。

#### 新技術：

以電容式近接感測器為基礎，使用一條線路作為近接感測器。隱密性高的線可綁在樹枝、岩石、或山泉水的洞口。只需一把剪

線鉗就能輕易調整觸發區域。在光亮的陽光和星光的夜間，功能完全不變。背景環境的熱溫不會產生任何影響。近接感測器會定期喚醒啟動，且只有當動物出現在目標區時才會啟動攝影機，以節省電池的電力。

比較兩種設置攝影機陷阱的方式，用電容感測技術設置攝影機陷阱，在某些情況下，勝過現有的設置方法。新方法在野生動物研究的領域，可找到市場利基。

## 設計流程簡介

免撰寫程式的設計流程有六個步驟

1. 以文字描述系統
2. 描繪系統模塊圖表
3. 定義狀態機器、轉換函式、以及真值表
4. 執行系統模擬
5. 測試實際系統
6. 微調CapSense

### 步驟1. 以文字描述系統

我們從上到下觀察設計流程。這意謂著以文字定義系統運作的程序，然後再視需求加入技術細節。以下是一段對系統的非技術性簡短描述。

*系統會持續監視任何動物。當偵測到動物時，攝影機會啟動並拍攝。在沒有偵測到動物時，攝影機就會關閉。為了避免拍攝到超出攝影機陷阱設定範圍以外的內容，當系統關閉時，偵測系統會解除警戒一段時間。*

這些描述是一個很好的起點，但在開始使用PSoC Express之前，還需更詳細的描述。以下是系統運作的技術描述：

1. 設定攝影機
2. 隱藏目標區的線路。例如：把線連結到樹枝或將絕緣線應融入環境之中。
3. 把感測器的線路與攝影機控制線連結到控制器機板。
4. 打開控制器機板的電源
5. 紅色LED閃爍30秒，讓研究人員有時間可以從系統前走開，避免觸發照相功能。在這段時間，攝影機的觸發器會關閉，攝影機也關閉。
6. 紅色LED停止閃爍，並維持關閉狀態

7. 攝影機的觸發器現在開啟，等候動物移到近接感測器附近
8. 當有動物觸發近接感測器，綠色LED就會開啟。攝影機啟動，並拍攝影像。
9. 當動物移出目標範圍，綠色LED就會關閉。攝影機在一般時間沒有輸入時，就會關閉。
10. 回到步驟7，重複執行步驟週期。

## 步驟2. 描繪系統模塊圖表

這個系統的模塊圖表是用PSoC Express來描繪，如圖1所示。圖表是透過一個圖形化拖曳來定義。PSoC Express有一個零組件函式庫，名為驅動器目錄，包含許多各種功能模塊以及溫度感測器、光線感測器、以及加速器。使用者視需要用滑鼠從目錄中把模塊拉到模塊圖表。為讓流程更簡化，這個設計範例中系統唯一的輸出就是紅色與綠色LED閃燈。使用者僅須從驅動目錄拉出兩個數位輸出端，就能將“攝影機開啟/關閉”與“快照”輸出功能加入至相機。

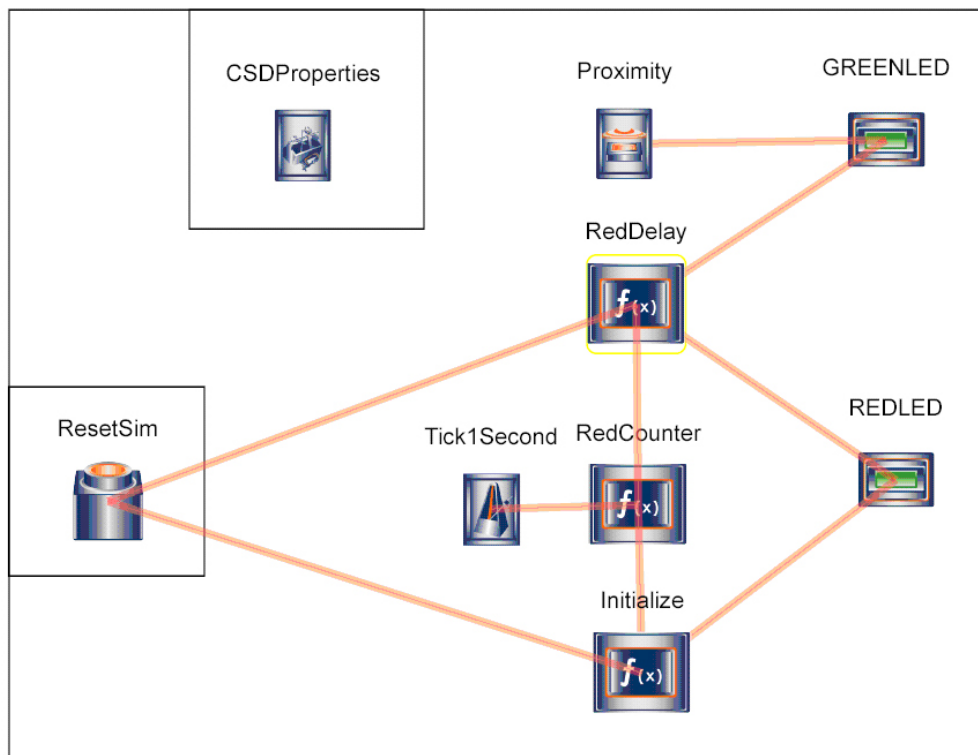


圖1. 解決方案模塊圖表

圖1的設計範例系統結合以下功能至模塊。

- Proximity –A近接感測器採用CapSense的CSD方法。
- 綠色LED閃燈 – 當近接感測器觸發時，綠色LED就會開啟
- 紅色LED閃燈 –在重置後30秒的Disarm週期內，紅色LED燈會閃爍
- Tick1Second –時基元件會以每秒一個脈衝的頻率，輸出一個脈衝
- RedCounter –計數器每秒遞增一個值。
- RedDelay –在30秒之後變更狀態的狀態機器
- Initialize –當系統被重置時，用來重置計數器的狀態機器

### 狀態3. 定義狀態機器、轉換函式、以及真值表

PSoC Express雖然不必撰寫程式碼，但必須為系統定義邏輯描述，這個作業是透過狀態機器、轉換函式、以及真值表來進行。

設計中有兩個簡單的狀態機器，其基本的結構相同。狀態圖是在PSoC Express中繪製，如圖2與圖3所示。轉換邏輯顯示在表中，已納入每個圖表。

系統中名為Initialize的狀態機器，讓模擬作業更加容易操作，名為ResetSim的重置開關控制這個程序。在圖2的轉換函式表中，當重置開關為ON狀態時，這個狀態機器會維持在State0狀態。其他時間狀態機器會切換至State1狀態，讓計數器的值能夠遞增。

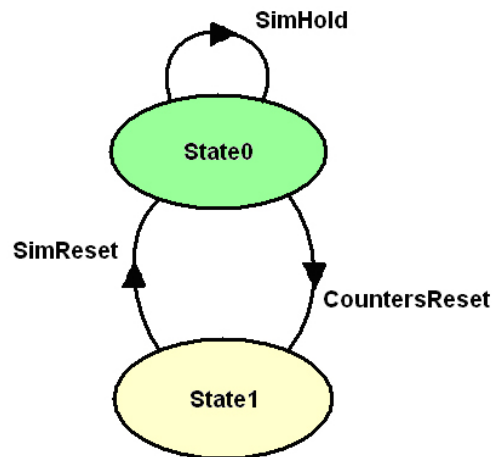


圖2.名為Initialize狀態機器的狀態圖

Transfer Function: Initialize State Machine

TRANSITION NAME	FROM STATE	TO STATE	EXPRESSION
CountersReset	State0	State1	ResetSim == ResetSim_Off
SimReset	State1	State0	ResetSim == ResetSim_On
SimHold	State0	State0	ResetSim == ResetSim_On

名為RedDelay的狀態機器被加入到系統，用來顯示系統重置後已經過30秒的時間。圖3的轉換函式表就是很好的說明。

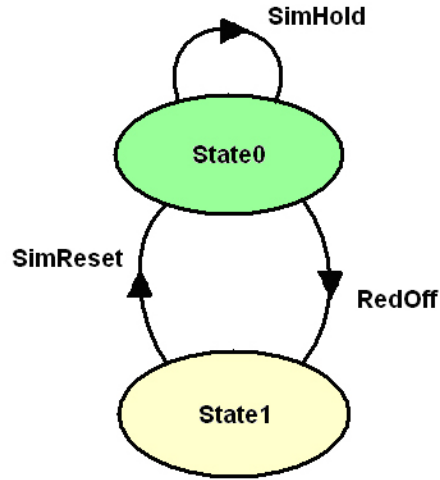


圖3. 名為RedDelay狀態機器的狀態圖

Transfer Function: RedDelay State Machine

TRANSITION NAME	FROM STATE	TO STATE	EXPRESSION
RedOff	State0	State1	RedCounter >= 30
SimReset	State1	State0	ResetSim == ResetSim_On
SimHold	State0	State0	ResetSim == ResetSim_On

設計中有三個真值表。這些真值表的特殊形態名為優先編碼器。編碼器有單一輸出端，其輸出的設定，由真值表中輸入的邏輯表示式來控制。這些邏輯規則依序進行測試，直到所有條件符合規則為止。這三個真值表用來讓計數器進行遞增，開啟與關閉LED燈，如圖4、圖5、與圖6所示。

Transfer Function: RedCounter Priority Encoder

EXPRESSION			
IF	Initialize_state == Initialize_state_State0	THEN	0
ELSEIF	RedCounter >= 60	THEN	RedCounter
ELSEIF	Tick1Second == Tick1Second_Triggered	THEN	RedCounter + 1

圖4. RedCounter定義

### Transfer Function: GREENLED Priority Encoder

EXPRESSION			
IF	(RedDelay_state==RedDelay_state__State0)	THEN	GREENLED__OFF
ELSEIF	(Proximity_Status > 0)	THEN	GREENLED__ON
ELSEIF	1	THEN	GREENLED__OFF

圖5. GreenLED 定義

### Transfer Function: REDLED Priority Encoder

EXPRESSION			
IF	Initialize_state==Initialize_state__State0	THEN	REDLED__ON
ELSEIF	RedDelay_state==RedDelay_state__State0	THEN	REDLED__ON
ELSEIF	RedDelay_state==RedDelay_state__State1	THEN	REDLED__OFF

圖6. RedLED 定義

#### 步驟4. 進行系統模擬

PSoC Express諸多強悍功能之一就是模擬。設計範例的模擬檢視圖如圖7所示。這個工具讓你能測試系統是否在會適當時機進行轉換。模擬功能讓你能執行what-if條件式實驗，不必對任何部份元件進行程式開發。若所有運作都符合預期，就可開始針對一部份元件撰寫程式，然後檢查系統在實際運作時的狀況。

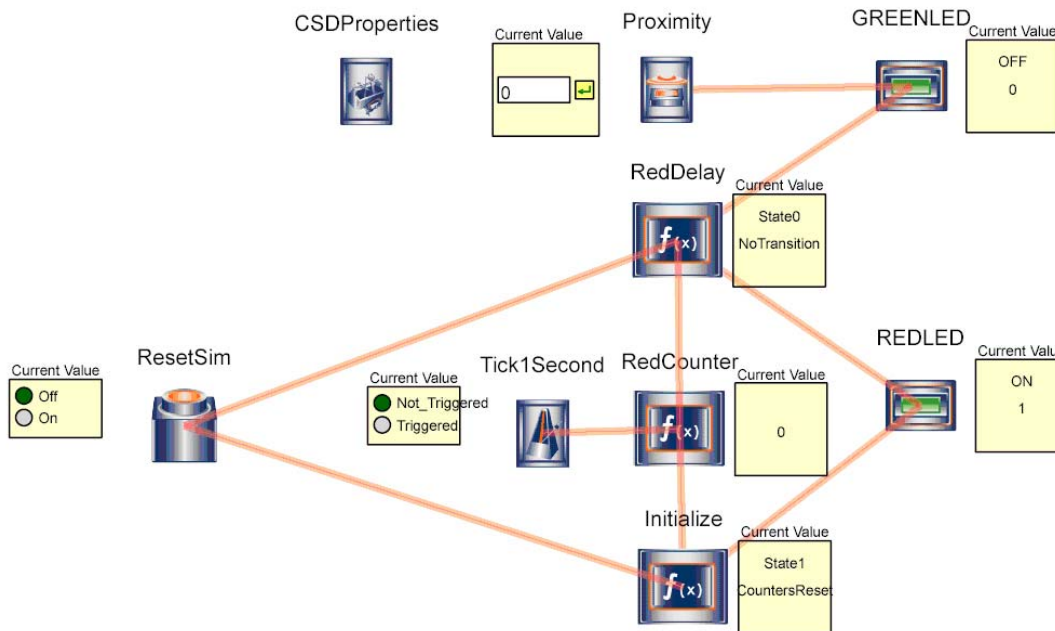


圖7. 對系統進行模擬

## 步驟5. 測試實際系統

FirstTouch評估套件的成本低廉，並整合至一個USB隨身碟中，如圖8與圖9所示。這個小巧的開發系統包含評估設計範例所需的所有工具。

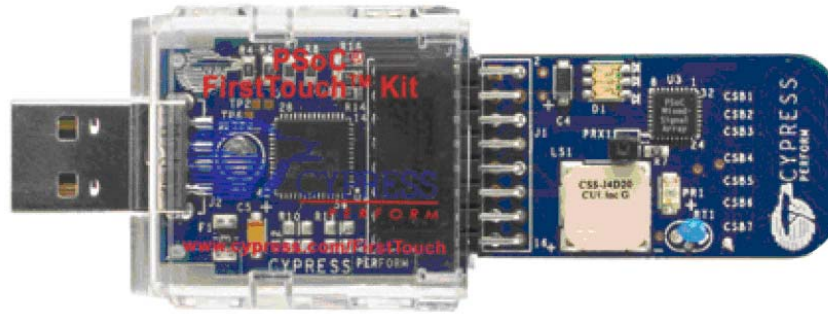


圖8. FirstTouch 套件為程式開發人員提供USB介面以及開發公板

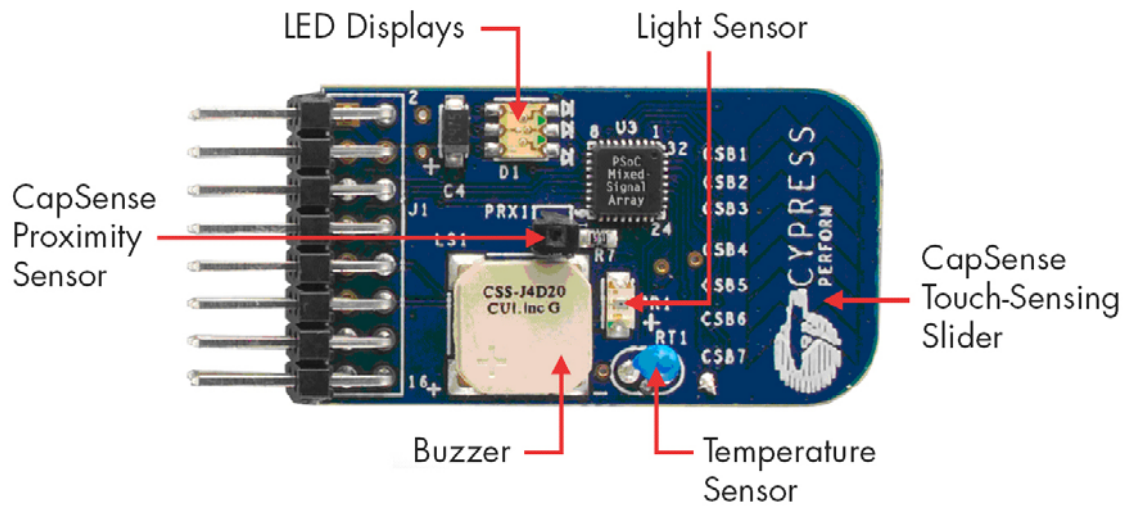


圖9. FirstTouch套件開發公板的詳圖，包括近接感測器線的連結點

FirstTouch套件的程式開發流程包括選擇正確的PSoC，也就是CY8C21434，然後為每項功能選擇正確的針腳，如圖10所示。專案的hex格式檔案在PSoC Express中建立，並編寫到FirstTouch套件。實際系統的行為會對應至模擬時預測的情況，以符合先前定義的系統需求。專案成功完成，新手僅須花數個小時就執行完畢。若是有經驗的PSoC Express使用者，設計流程甚至會更快。

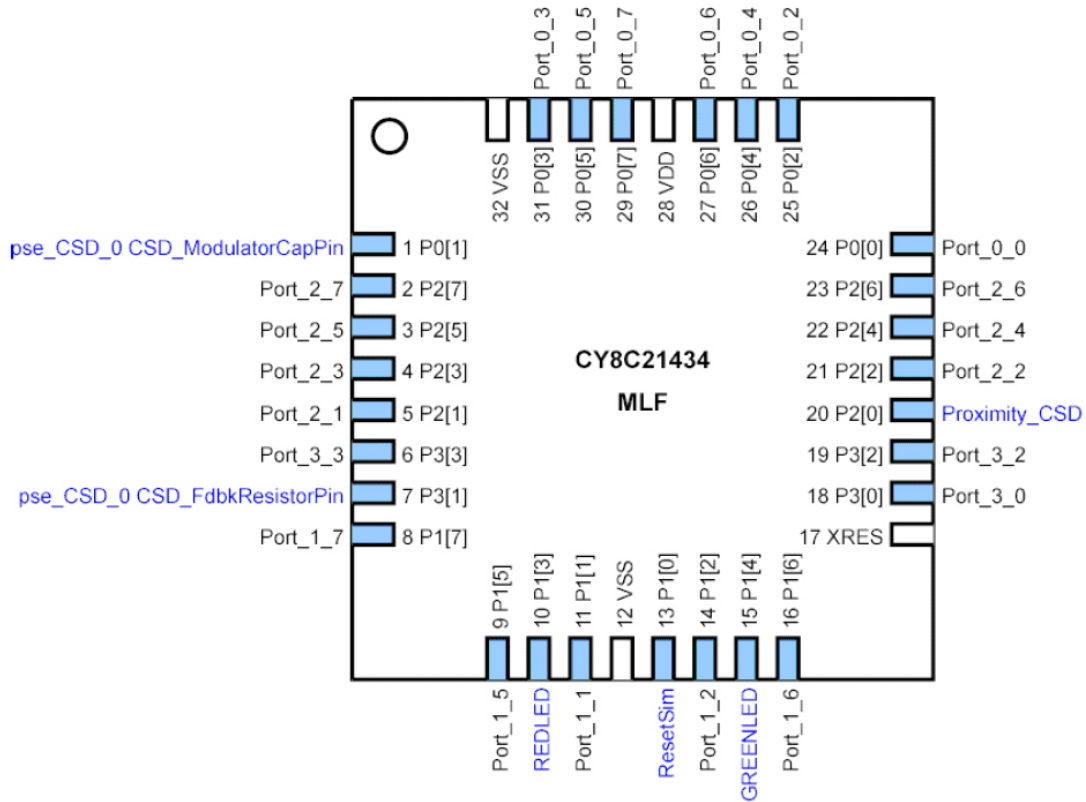


圖10. FirstTouch開發公板專案的針腳配置

### 步驟6. 對CapSense進行微調

這項模擬流程中沒有涵蓋的一項功能就是CapSense感測模塊的微調。CSD與近接屬性設定，在第一Touch套件中運作得相當好，如圖11與12所示。

Name	CSDProperties
Driver	Properties - CSD
Cypress Certified	yes
Version	1.1.0.2
NoiseThreshold	10
NegativeNoiseThres	20
BaselineUpdateThre	200
Hysteresis	0
Debounce	3
LowBaselineReset	50
Sensors Autoreset	Enabled

圖11. CSD Properties的各項屬性



Name	Proximity
Driver	Proximity - CSD
Cypress Certified	yes
Version	1.0.0.4
Property Editor	C:\PROGRAM FILES\CYPRESS
DetectionThreshold	1
Hysteresis	15
Ref Value	0
Scan Speed	Slow
Scanning Resolution	16
Expose Tuning Values	Yes

圖12. 近接感測器的各項屬性

### 未來發展

試想當研究人員想拍的動物靠近樹枝，而觸發裝置正好裝在樹枝上，因動物觸碰到樹枝而沒有觸發。像是蛇或蝸牛爬在樹枝上而觸發了近接感測器，但研究人員並不想研究這些生物。為了捕捉接近感測器但沒有觸碰到感測器的動物，研究人員只須在棒子上加裝觸碰感測器。要在PSoC Express進行這項變更，須在驅動器目標中選擇CapSense按鈕，然後變更開啟綠色LED的邏輯(當 Proximity\_status >0 且Button\_status <1時啟動)。這樣就能建構出一個智慧型捕鼠陷阱!

Cypress Semiconductor  
 198 Champion Court  
 San Jose, CA 95134-1709  
 Phone: 408-943-2600  
 Fax: 408-943-4730  
<http://www.cypress.com>

© Cypress Semiconductor Corporation, 2007. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

PSoC Designer™, Programmable System-on-Chip™, and PSoC Express™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.