

Tronlong®

SOM-TL6678

核心板规格书



广州创龙电子科技有限公司

© 2013 Guangzhou Tronlong Electronic Technology Co.,Ltd.

Revision History

Draft Date	Revision No.	Description
2018/04/08	V1.3	1. 更新封面、产品图以及硬件框图。 2. 修改运用领域。
2018/02/22	V1.2	1. 修改附录 A 开发例程。 2. 修改核心板工作环境。
2017/05/03	V1.1	1.更新为 A3 版。
2016/03/16	V1.0	1.初始版本。

目 录

1 核心板简介.....	4
2 典型运用领域.....	6
3 软硬件参数.....	6
4 开发资料.....	8
5 电气特性.....	9
6 机械尺寸图.....	9
7 产品订购型号.....	10
8 技术支持.....	11
9 增值服务.....	11
更多帮助.....	12
附录 A 开发例程.....	13

1 核心板简介

- 基于 TI KeyStone C66x 多核定点/浮点 TMS320C6678 DSP，集成了 8 个 C66x 核，支持高性能信号处理应用；
- 每核心主频 1.0/1.25GHz，单核可高达 40GMACS 和 20GFLOPS，每核心 32KByte L1P、32KByte L1D、512KByte L2，4MByte 多核共享内存，8192 个多用途硬件队列，支持 DMA 传输；
- 支持双千兆网口，带有由 1 个数据包加速器和 1 个安全加速器组成的网络协处理器；
- 支持 PCIe、SRIO、HyperLink、EMIF16 等多种高速接口，同时支持 I2C、SPI、UART 等常见接口；
- 连接稳定可靠，80mm*58mm，体积极小的 TMS320C6678 核心板，采用工业级高速 B 2B 连接器，高速数据接口使用高速连接器，保证信号完整性；
- 提供丰富的开发例程，入门简单，支持裸机和 SYS/BIOS 操作系统。

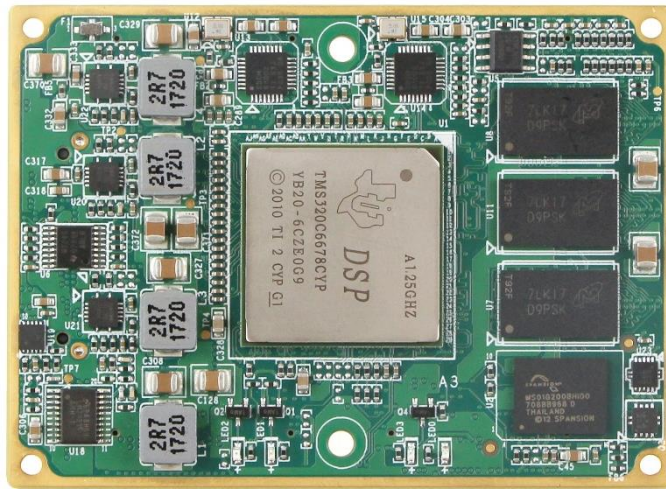


图 1 核心板正面图

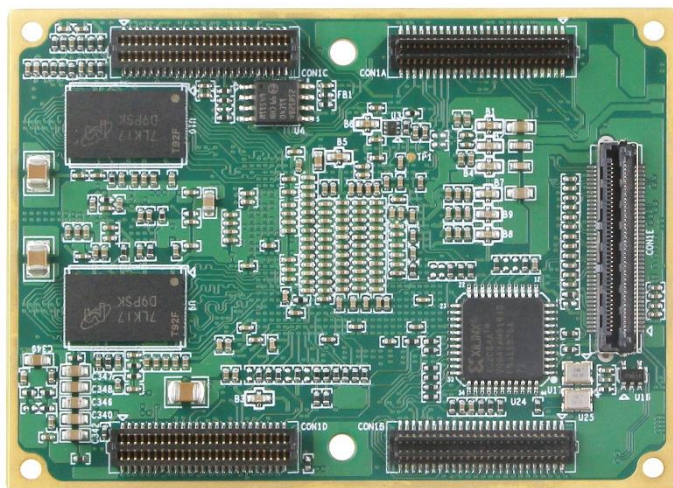


图 2 核心板背面图



图 3 核心板斜视图

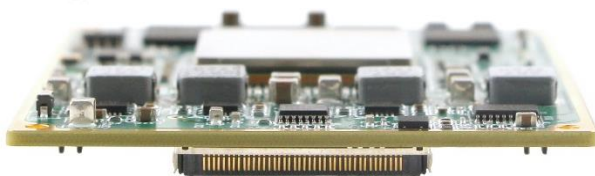


图 4 核心板侧视图

由广州创龙自主研发的 SOM-TL6678 是一款基于 TI KeyStone 系列多核架构的定点/浮点 TMS320C66x 高端 DSP 核心板，采用沉金无铅工艺的 12 层板设计，专业的 PCB Layout 保证信号完整性的同时，经过严格的质量控制，满足多种环境应用。

SOM-TL6678 引出 CPU 全部资源信号引脚，二次开发极其容易，客户只需要专注上层运用，降低了开发难度和时间成本，让产品快速上市，及时抢占市场先机。

不仅提供丰富的 Demo 程序，还提供 DSP 核间通信开发教程，全面的技术支持，协助客户进行底板设计和调试以及多核软件开发。

2 典型运用领域

- ✓ CT 扫描仪
- ✓ 机器视觉
- ✓ X 射线：行李扫描仪
- ✓ 信号测量：信号分析仪
- ✓ 雷达/声纳
- ✓ 可编程多核视频处理器
- ✓ 无线通信测试仪
- ✓ 视频分析服务器
- ✓ 超声波系统
- ✓ 软件无线电
- ✓ 高速数据采集和生成

3 软硬件参数

硬件框图

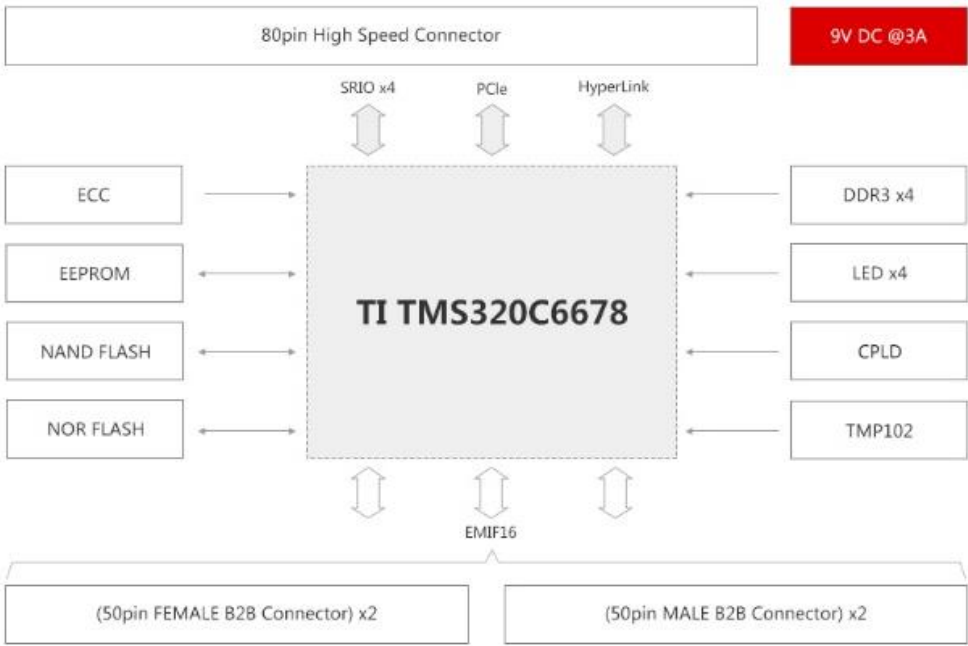


图 5 核心板硬件框图

硬件参数

表 1

CPU	TMS320C6678, 8 核 C66x, 主频 1.0/1.25GHz
ROM	128MByte NAND FLASH
	16MByte SPI NOR FLASH
RAM	1/2GByte DDR3
EEPROM	1Mbit
ECC	256/512MByte DDR3
SENSOR	1x TMP102AIDRLT, 核心板温度传感器, I2C 接口
LED	1x 供电指示灯
	1x CPLD 状态灯
	2x 用户指示灯
B2B Connector	2x 50pin 公座 B2B, 2x 50pin 母座 B2B, 间距 0.8mm, 合高 5.0mm; 1x 80pin 高速 B2B 连接器, 间距 0.5mm, 合高 5.0mm, 共 280pin, 信号速率最高可达 10GBaud
硬件资源	1x SRIO, 四端口四通道, 每通道最高通信速率 5GBaud

1x PCIe Gen2, 单端口双通道, 每通道最高通信速率 5GBaud
2x Ethernet, 10/100/1000M
1x EMIF16, 16bit
1x HyperLink, 最高通信速率 50GBaud, KeyStone 处理器间互连的理想接口
2x TSIP
1x UART
1x I2C
1x SPI
16x TIMER
16x GPIO
1x JTAG
1x BOOTMODE, 13bit

软件参数

表 2

DSP 端软件支持	裸机、SYS/BIOS 操作系统
CCS 版本号	CCS5.5
软件开发套件提供	MCSDK

4 开发资料

- (1) 提供核心板引脚定义、可编辑底板原理图、可编辑底板 PCB、芯片 Datasheet, 缩短硬件设计周期;
- (2) 提供丰富的 Demo 程序, 包含 DSP 多核通信教程, 完美解决多核开发瓶颈;
- (3) 提供完整的平台开发包、入门教程, 节省软件整理时间, 上手容易;

部分开发例程详见附录 A, 开发例程主要包括:

➤ 裸机开发例程

创龙

- SYS/BIOS 开发例程
- 多核开发例程

5 电气特性

核心板工作环境

表 3

环境参数	最小值	典型值	最大值
工业级温度	-40°C	/	85°C
工作电压	/	9V	/

核心版功耗

表 4

	典型值电压	典型值电流	典型值功耗
核心板	9.17V	961.6mA	8.82W

备注：功耗测试基于广州创龙 TL6678-EasyEVM 开发板进行。

6 机械尺寸图

表 5

PCB 尺寸	80mm*58mm
安装孔数量	4 个
散热器安装孔数量	2 个

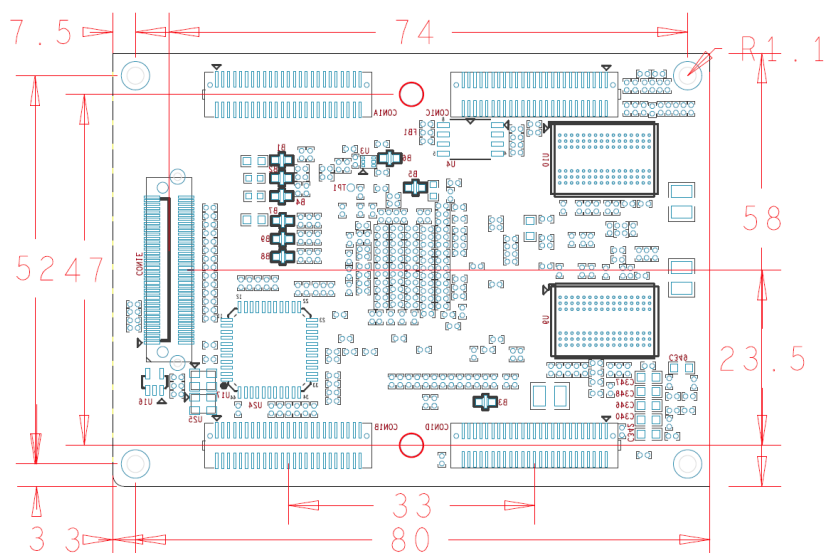


图 6 核心板机械尺寸图

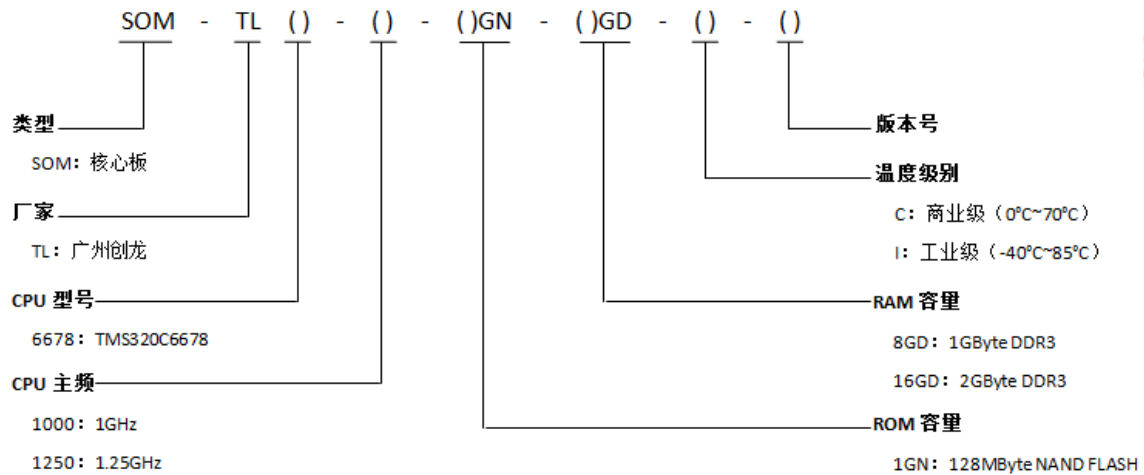
7 产品订购型号

表 6

型号	CPU 主频	NAND FLASH	DDR3	温度级别
SOM-TL6678-1000-1GN8GD-I	1.0GHz/核	128MByte	1GByte	工业级
SOM-TL6678-1000-1GN16GD-I	1.0GHz/核	128MByte	2GByte	工业级
SOM-TL6678-1250-1GN8GD-I	1.25GHz/核	128MByte	1GByte	工业级
SOM-TL6678-1250-1GN16GD-I	1.25GHz/核	128MByte	2GByte	工业级

备注：标配为 SOM-TL6678-1000-1GN8GD-I，其他型号请与相关销售人员联系。

型号参数解释



8 技术支持

- (1) 协助底板设计和测试，减少硬件设计失误；
- (2) 协助解决按照用户手册操作出现的异常问题；
- (3) 协助产品故障判定；
- (4) 协助正确编译与运行所提供的源代码；
- (5) 协助进行产品二次开发；
- (6) 提供长期的售后服务。

9 增值服务

- 主板定制设计
- 核心板定制设计
- 嵌入式软件开发
- 项目合作开发
- 技术培训

更多帮助

销售邮箱: sales@tronlong.com

技术邮箱: support@tronlong.com

创龙总机: 020-8998-6280

技术热线: 020-3893-9734

创龙官网: www.tronlong.com

技术论坛: www.51ele.net

线上商城: <https://tronlong.taobao.com>

TMS320C665x、TMS320C6678 交流群: 79635273、332643352

TI 中文论坛: <http://www.deyisupport.com/>

TI 英文论坛: <http://e2e.ti.com/>

TI 官网: www.ti.com

TI WIKI: <http://processors.wiki.ti.com/>

附录 A 开发例程

表 7

算法开发例程	
例程	功能
FFT_Real	快速傅里叶变换/逆变换
FFT_Real_Benchmark	快速傅里叶变换/逆变换（打开/关闭缓存速度对比）
FIR	有限长单位冲激响应滤波器
IIR	无限脉冲响应数字滤波器
DCT	图像离散余弦变换
RGB2Gray	RGB24 图像转灰度
Canny	边缘检测
HIST	灰度图像直方图
Threshold	灰度图像二值化
Rotate	图像旋转
Zoom	图像缩放
ImageReverse	图像反色
InteEqualize	直方图均衡化
LinerTrans	灰度图像线性变换
MATH	数学函数库
Matrix	矩阵运算
Algorithm_LSB_Hide	图片添加水印
Algorithm_Plate_Recognition	车牌识别

表 8

SYS/BIOS 开发例程	
例程	功能

CLOCK	时钟
Task	任务
Task_MUTEX	抢占式多任务
Task_STATIC	静态创建任务
Timer_C6678	定时器（专用）
Timer	定时器（通用）
Timer_C6678_Runtime	定时器（动态创建）
HWI_C66x	硬件中断（HWI 设备专用组件）
HWI_C66x_Hook	硬件中断（HWI 挂钩函数）
HWI_C66x_Nest	硬件中断（HWI 中断嵌套）
HWI_Runtime	硬件中断（HWI）
HWI_Runtime_Post_SWI	硬件中断（HWI 发布软件中断）
HWI_Runtime_Post_Task	硬件中断（HWI 触发任务）
MEMORY	内存分配
SWI	软件中断（静态配置）
SWI_Runtime	软件中断（SWI）
SWI_Runtime_Post_Conditionally_andn	软件中断（有条件触发 ANDN）
SWI_Runtime_Post_Conditionally_dec	软件中断（有条件触发 DEC）
SWI_Runtime_Post_Unconditionally_or	软件中断（无条件触发 OR）
Timestamp	时间戳（通用）
Timestamp_C6678	时间戳（专用）
UART_POLL	UART0 串口查询收发
UART_INT_FIFO	UART0 串口中断收发
UART_INT	UART 串口中断收发
EDMA3	EDMA3 一维数据传输
PCle	PCle 板间通信
SRIO	SRIO 板间通信

SRIO_4x_FPGA2DSP	FPGA 与 DSP 的 SRIO 通信测试
NDK_TCP	TCP 服务器
NDK_TCP_Client	TCP 客户端
NDK_TCP_Benchmark	TCP 发送/接收速度测试
NDK_UDP	UDP 通信
NDK_Telnet	Telnet 协议
NDK_WebServer	网络 Web 服务器
NDK_Runtime	网络 Web 服务器（支持串口输入 IP）
NDK_RawSocket	以太网数据链路层通信
Board_C6678	开发板全基本功能测试
NDK_UIA	基于网络传输的系统分析
NDK_DualPort_Runtime	基于 NDK 的双网口 Web 服务器（不支持串口输入 IP）

表 9

裸机开发例程	
例程	功能
GPIO_LED	GPIO 输出（LED 灯）
GPIO_KEY	GPIO 输入（按键中断）
UART_POLL	UART1 串口查询收发
TimerLED	定时器调整 LED 控制脚频率
Fan	对散热风扇转速进行控制
NonOS_MPAX	访问相同的逻辑地址
GPIO_LED_C++	GPIO 输出（LED 灯）
GPIO_LED_Assembly	GPIO 输出（标准汇编）
GPIO_LED_LinearAssembly	GPIO 输出（线性汇编）

表 10

MultiCore 多核开发例程	
例程	功能
OpenMP_Hello	OpenMP 的测试
OpenMP_Hello_SYSBIOS	基于 SYSBIOS 的 OpenMP 测试
OpenMP_Matrix-Vector_Multiplication	基于 OpenMP 的矩阵-向量乘法
OpenMP_RGB2Gray	基于 OpenMP 的 RGB24 图像转灰度
OpenMP_MPAX	访问相同的逻辑地址
MultiCore_IPC_MessageQ	MessageQ 模块通讯测试
MultiCore_IPC_Notify	Notify 模块通信测试
MultiCore_IPC_SharedRegion	SharedRegion 模块通信测试
MultiCore_IPC_RGB2Gray	基于 IPC 的 RGB24 图像转灰度
MultiCore_DualImage	多核多镜像通信测试
MultiCore_DualImage_SYSBIOS	多核多镜像 SYSBIOS 通信测试
MultiCore_SingleImage	多核单镜像通信测试
MultiCore_SingleImage_Semaphore2	多核单镜像通信测试
MultiCore_SingleImage2_Semaphore2_SYSBIOS	多核单镜像 SYSBIOS 通信测试

表 11

基于广州创龙编写的 RTSC 组件的例程	
例程	功能
RTSC_Fan	对散热风扇转速进行控制
RTSC_FFT_Real	快速傅里叶变换/逆变换
RTSC_UART_POLL	UART 串口查询收发
RTSC_I2C_TempSensor	IIC 总线温度传感器测试
RTSC_SysMin	SysMin 组件的输出调试演示
RTSC_SysStd	SysStd 组件的输出调试演示
RTSC_LoggerBuf	日志输出到缓冲区的演示例程

RTSC_LoggerStd	实时输出日志信息的演示例程
RTSC_Benchmark	测量代码性能方法
RTSC_UART_INTRRUPT	UART 串口中断收发
RTSC_LED	LED 测试
RTSC_KEY	按键测试
RTSC_LoggerUART	日志输出到缓冲区的演示例程