**CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI**

* 1. **Giới thiệu chương**

Cùng với sự phát triển không ngừng của khoa học công nghệ, công nghiệp ô tô cũng đã và đang được hoàn thiện nhằm mục đích đem đến cho người dùng những trải nghiệm ngày càng thoải mái và tiện nghi hơn. Sự ra đời của các công nghệ, thiết bị hiện đại như cảnh báo chệch làn đường, cảnh báo tiền va chạm, hệ thống hỗ trợ phanh và điều khiển động cơ bánh lái…đã mang lại cho người dùng những chuyến hành trình ngày càng thú vị. Tuy nhiên, để áp dụng những công nghệ mới này lên trên một sản phẩm xe ô tô thực tế để đưa ra ngoài thị trường thì từ khâu kiểm thử, vận hành cho đến phát triển các ứng dụng về sau thì đây cũng là một khó khăn đối với các nhà nghiên cứu, cũng như sản xuất ô tô công nghiệp. Việc sử dụng cả hệ thống thực tế chỉ để kiểm thử với một ECU (Electronic Control Unit) mới nhằm hướng tới sự tương thích giữa các ECU với nhau có thể gây ra sự phức tạp cho toàn bộ hệ thống CAN BUS (Controller Area Network ).

Chính vì vậy, việc xây dựng nên một phần mềm có thể mô phỏng hoạt động như một hệ thống hoàn chỉnh nhằm thay thế các ECU thực tế là thật sự cần thiết. Nó sẽ giúp cho việc phát triển thêm ứng dụng đơn giản hơn, với độ tin cậy và hiệu quả cao.

* 1. **Tổng quan về hệ thống**

Đây là sơ đồ khối mô hình kết nối khi sử dụng , sơ đồ khối mô tả chung của đề tài.



*Hình 1.1 : Sơ đồ khối mô hình kết nối khi sử dụng*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | | |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | *Hình 1.2 : Sơ đồ khối mô tả chung hệ thống CAN SIMULATOR*. |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Đề tài được thực hiện chủ yếu dựa trên việc lập trình ngôn ngữ C/C++, sử dụng kit TM4C123G của hãng Texas Instruments và một số phần mềm hỗ trợ khác như CCS, Qt Creator,....

* 1. Các thành phần trong CAN Simulator

Hệ thống gồm có 3 phần chính:

1.3.1 – CAN Software PC

Là một phần mềm độc lập, có nhiệm vụ xử lí việc truyền nhận dữ liệu cũng như phân tích và hiển thị nội dung của các thông điệp, mà định dạng của các thông điệp đó được lấy từ D atabase edit software.

Chức năng:

* Cấu hình hệ thống, cấu hình được cổng CAN nào đang hoạt động, tốc độ Baund

Load được database vào test model

* Add/remove được sơ đồ khối kết nối đồng thời cập nhật cấu hình và lưu lại.
* Add/remove được các message vào trong IG.
* Trong IG (interactive generator) có thể sửa được dạng sóng của các signal: hình sin, răng cưa, hình chữ nhật,…
* Trong IG có thể setting cycle của từng message.
* Trong lúc đang simulate có thể edit nội dung message ở dạng raw data hoặc physical value.
* Trace window hiển thị các thuộc tính của message: ID, length, direction, data, signal value.

1.3.2. PC Driver - Protocol

PC Driver có nhiệm vụ giao tiếp giữa hardware và software .Khi một device được kết nối với PC, driver nhận nhiệm vụ tạo ra vùng nhớ đệm thông qua các file descriptor. Khi ta muốn truyền một thông điệp từ PC xuống Hardware thì Driver sẽ nhận biết được thông điệp, sau đó ghi vào file descriptor tương ứng. Hardware sẽ đọc dữ liệu từ file descriptor và thực hiện nội dung thông điệp. Thông điệp phản hồi sẽ được truyền theo chiều ngược lại đến Software.

* Nhận biết được thiết bị connect, disconnect thông qua cổng USB, đưa ra thông tin, trạng thái của thiết bị.
* CAN software có thể đóng/ mở thiết bị thông qua device node.
* CAN software có thể cấu hình baurd rate thông qua file descriptor.

CAN software có thể gửi/nhận message thông qua file descriptor

1.3.3. CAN Simulator Device

Cấu hình 2 giao tiếp CAN và USB để có thể nhận và gửi các message.

* Nhiệm vụ chính:
* Giao tiếp USB nhận và gởi message để giao tiếp giữa board và PC
* Giao Tiếp CAN
* Nhận CAN Frame từ CAN BUS và gởi dữ liệu thu được từ CAN Frame lên PC.
* Kiểm tra thiết bị đã được kết nối với CAN BUS hay chưa, trả kết quả lại cho PC.
* Cấu hình được tốc độ của thiết bị để có thể kết nối được với BUS CAN bất kỳ.
  1. Kết luận chương

Trong phần này chúng ta đã tìm hiểu sơ bộ về đề tài và các thành phần chính trong đề tài. Tuy nhiên vì đề tài rất lớn với nhiều nhóm sinh viên cùng thực hiện các phần nhỏ liên quan, nên sẽ có nhiều nhóm nhỏ thực hiện các phần khác nhau. Vì vậy, trong các phần tiếp theo chúng ta sẽ tìm hiểu cụ thể hơn về CAN Simulator Device.

**Chương 2: CAN SIMULATOR DEVICE**

**2.1. Giới thiệu chung**

Chức năng chính của CAN Simulator Device là tạo một node CAN có thể tùy chỉnh được tốc độ của node để có thể kết nối được với một CAN BUS bất kỳ để lấy các CAN\_Frame trong CAN BUS và gởi lên PC.

Để thực hiện các chức năng đó chúng ta dung giao thức USB và CAN. Board DK-TM4C123G hỗ trợ đầy đủ các yêu cầu trên và chúng ta sẽ tìm hiểu kỹ về giao thức CAN, USB trong phần này.

**2.2. Lý thuyết về phần cứng**

2.2.1. Giao thức USB – USB Communications Device Class (USB CDC)

2.2.1.1. Tổng quan về giao thức USB

USB (Universal Serial Bus) là một chuẩn kết nối tuần tự đa dụng trong máy tính. USB sử dụng để kết nối các thiết bị ngoại vi với máy tính, chúng thường được thiết kế dưới dạng các đầu cắm cho các thiết bị tuân theo chuẩn cắm-và-chạy (plug and play) mà với tính năng cắm nóng thiết bị (nối và ngắt các thiết bị không cần phải khởi động lại hệ thống).

USB là một giao thức nối tiếp tốc độ cao, nó có thể cung cấp điện cho các thiết bị kết nối với nó. Một bus USB sử dụng trường địa chỉ 7-bit để đánh địa chỉ cho các thiết bị trong BUS, lưu ý rằng địa chỉ 0 không được sử dụng vì nó có một mục đích đặc biệt.

Giao thức USB có hai mode hoạt động chính là host mode và device mode:

* USB Host : thiết bị đóng vai trò điều khiển toàn bộ mạng USB (có thể lên tới tối đa 126 thiết bị). Ví dụ như trên máy tính, USB Host được gắn trên mainboard. Để giao tiếp và điều khiển các USB device, USB Host controller cần được thiết kế tích hợp với USB RootHub (Hub mức cao nhất). Vai trò của thiết bị USB Host:
* Trao đổi dữ liệu với các USB device.
* Quản lý USB BUS
* USB Device: là các thiết bị đóng vai trò như các slave giao tiếp với USB Host. Xin lưu ý một điều hết sức quan trọng đó là các thiết bị này hoàn toàn đóng vai trò bị động, không bao giờ được tự ý gửi gói tin lên USB Host hay gửi gói tin giữa các USB Device với nhau, tất cả đều phải thông qua quá trình điều phối của USB Host. Các bạn sẽ hiểu cơ chế này rõ hơn trong phần truyền thông của chuẩn USB. Chức năng của thiết bị USB Device:
* Trao đổi dữ liệu với USB Host
* Phát hiện gói tin hay yêu cầu từ USB Host

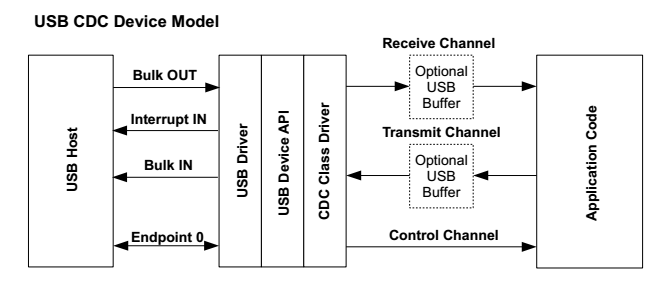
2.2.1.2. Các khái niệm cơ bản

* Endpoint: giống như là môt phần địa chỉ duy nhất của thiết bị, nó là nguồn thông tin trong dòng liên lạc giữa Host và thiết bị. Về cơ bản các thiết bị được thiết kế với tối ta 16 Endpoint, được phần loại thành hai loại:
* Các Endpoint truyền dữ liệu từ USB Device tới USB Host là endpoint IN.
* Các Endoint truyền dữ liệu từ USB Host tới USB Device là endpoint OUT.
* Pipes: là sự kết hợp giữa endpoint ở thiết bị với controller trên Host. Host có thể tạo các pipes mới hay hủy các pipe không cần thiết luân phiên theo yêu cầu của giao tiếp với thiết bị.
* Các chế độ truyền: chuẩn USB cung cấp cho chúng ta bốn chế độ truyền, tùy và mục đích và yêu cầu sử dụng mà người sử dụng chế độ phù hợp
* Truyền điều khiển (Control transfer): là chế độ truyền được tất cả các thiết bị USB hỗ trợ để truyền các thông tin điều khiển với tốc độ tương đối chậm.
* Truyền ngắt (Interrupt transfer): sử dụng cho các thiết bị cần truyền một lượng dữ liệu nhỏ, tuần hoàn theo thời gian ví dụ như chuột, bàn phím. Khi đó, ví dụ cứ 10s một lần USB Host sẽ gửi request xuống và USB Device sẽ trả dữ liệu về cho USB Host (với trường hợp Interrupt In Endpoint).
* Truyền theo khối (Bulk transfer): sử dụng cho các thiết bị cần truyền một lượng dữ liệu lớn, yêu cầu độ chính xác tuyệt đối, không có ràng buộc quá chặt chẽ về thời gian thực ví dụ như thẻ nhớ USB, máy in. Cái này tương tự như giao thức TCP trong mạng Ethernet
* Truyền đẳng thời (Isochronos transfer): sử dụng cho các thiết bị cần truyền một lượng dữ liệu lớn với tốc độ rất nhanh, đảm bảo ràng buộc về thời gian thực tuy nhiên chấp nhận hy sinh độ chính xác ở một mức nhất định như các thiết bị nghe nhạc, xem phim kết nối theo chuẩn USB

2.2.1.3. USB Communications Device Class

USB Communications Device Class là một lớp thiết bị trong USB Device, nó cho phép các ứng dụng của thiết bị được nhận thông qua cổng serial ảo trong hệ thống. Driver cung cấp hai kênh truyền, một truyền và một nhận. Các kênh truyền kết hợp với bộ đệm cung cấp cho chúng ta các giao diện để đọc và ghi dữ liệu với host USB.

Dữ liệu trong USB Communications Device Class sử dụng bốn Endpoint để truyền và nhận giữ liệu. Hai Bulk Endpoint để truyền và nhận dữ liệu từ host và Interrupt IN Endpoint để dùng để báo hiệu các lỗi như lỗi khung truyền, lỗi Parity bit được thiết bị phát hiện, cho Host. Endpoint 0 mang các yêu cầu của chuẩn USB và các yêu cầu cụ thể của CDC thông qua control chanel callback.



*Hình 2 mô hình của USB CDC*

2.2.2. Giao thức CAN

2.2.2.1. Tổng quan về giao tiếp CAN

Controller Area Network (CAN) là một giao thức truyền thông bus nối tiếp phát triển bởi Bosch (một nhà sản xuất thiết bị điện tại Đức) vào đầu những năm 1980. Sau đó, CAN đã được chuẩn hóa theo tiêu chuẩn ISO-11898 ,ISO-11519 và nó được sử dụng như là giao thức chuẩn để giao tiếp kết nối mạng trong ngành công nghiệp ôtô.

CAN được tổ chức International Standization Organization (ISO) định nghĩa là một chuẩn truyền thông nối tiếp, chuyên cho ngành công nghiệp Ô tô, nhằm thay thế cho việc đi dây phức tạp bằng BUS hai dây.

CAN được thiết kế là một mạng đa chủ, mỗi thiết bị đều có thể gởi và nhận tín hiệu. Không giống như các loại mạng truyền thống, CAN không gởi một gói tin lớn từ điểm này đến điểm kia mà nó chia mà nó gởi nhiều gói tin nhỏ theo dạng broadcast đến tất cả các Node trong BUS. Các Node trên BUS bắt các gói tin và dựa và trường ID trong gói tin nhận được để xử lý hay loại bỏ gói tin.

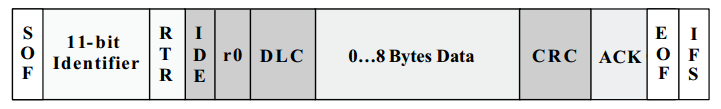
CAN được dựa trên giao thức CSMA / CD+AMP (Carrier-Sense Multiple Access: Đa truy cập có tránh xung đột / Collision Detection Arbitration on Message Priority: Giám sát phát hiện xung đột dựa trên thông điệp ưu tiên) , nó tương tự như các giao thức được sử dụng trong mạng Ethernet LAN.

Khi mạng Ethernet phát hiện một vụ xung đột, các nút đang gửi dữ liệu chỉ đơn giản là dừng truyền và đợi một khoảng thời gian ngẫu nhiên trước khi cố gắng để gửi một lần nữa. Tuy nhiên, đối với giao thức CAN có một chúc khác biệt đó là các vấn đề xung đột được giải quyết bằng cách sử dụng các nguyên tắc của trọng tài, nơi chỉ có các nút ưu tiên cao nhất được trao quyền để gửi dữ liệu của nó.

2.2.2.2. Standard CAN – Extended CAN

Về cơ bản hiện nay, có hai loại giao thức CAN chính là Standard CAN và Extended CAN. Chúng ta sẽ tìm hiểu kỹ hơn về hai loại giao thức này trong phần này:

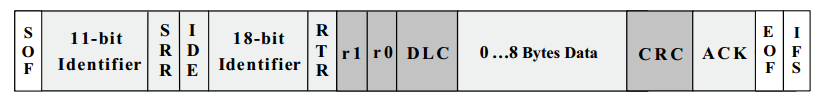
2.2.2.2.1. Standard CAN



*Standard CAN Frame*

|  |  |  |
| --- | --- | --- |
| Tên | Độ dài | Mục đích |
| SOF | 1 bit | Bit bắt đầu của một frame |
| Identifier(ID) | 11 bit | Chứa ID của message |
| RTR | 1 bit | Phân biệt data frame hay remote frame |
| IDE | 1 | Phân biệt loại extended hay standard |
| R0 | 1 | Bit dành riêng.(chỉ dành để sủa đổi tiêu chuẩn trong tương lai) |
| DLC | 4 | Độ dài của trường data |
| Data | 0-64 bit | Dữ liệu của message |
| CRC | 16 bit | Mã kiểu tra lỗi CRC |
| ACK | 2 bit |  |
| EOF | 1 bit | Kết thúc của Frame |
| IFS | 7 bit | Interframe Space |

2.2.2.2.2. Extended CAN



*Extended CAN Frame*

Khác biệt giữa Standard CAN và Extended CAN là trường ID 29 bit và một số trường khác.

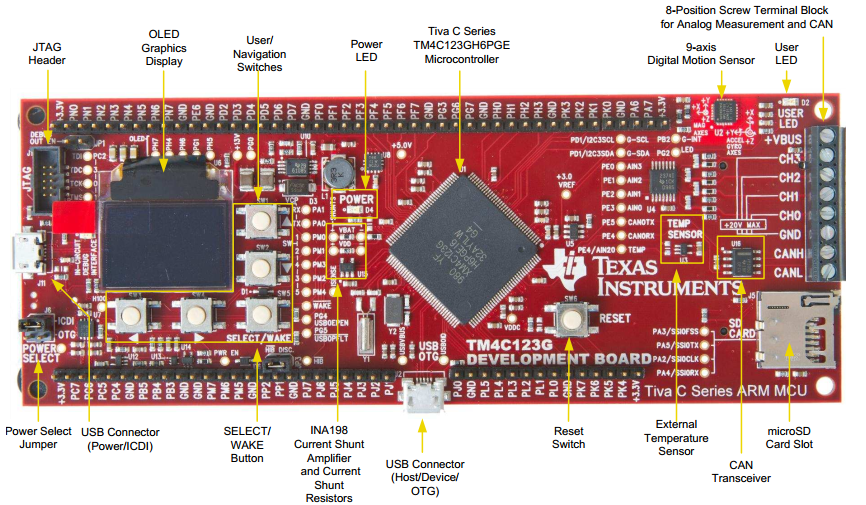
**2.3. Giới thiệu về phần cứng**

2.3.1. Giới thiệu board DK-TM4C123G

Kit DK-TM4C123G là dòng Kit phát triển do TI sản xuất, chuyên hỗ trợ cho các ứng dựng sử dụng giao thức USB và CAN.

DK-TM4C123G là dòng Kit mạnh mẽ và được phát triển sử dụng dong chip Tiva C Series TM4C123G ARM® Cortex™-M4-based. Kit được thiết kế đặc biệt với chip TM4C123G và được tích hợp các giao thức như:

* Giao thức USB 2.0 (OTG, Host, Device).
* Giao thức Controller Area Network (CAN)
* 12-bit Analog-to-Digital Converter (ADC)
* Thời gian thực (RTC) cùng pin cấp nguồn cho module RTC riêng biệt
* …

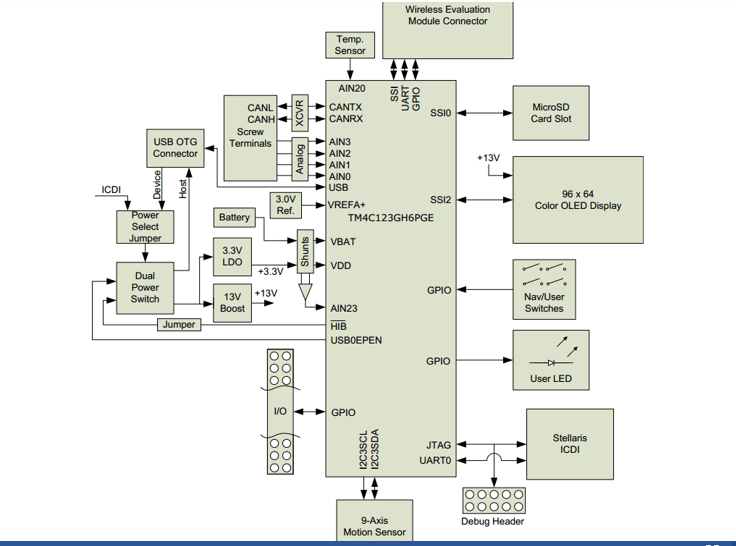


*Hình 2.: DK-TM4C123G do TI sản xuất*

2.3.2. Thông số cơ bản về Kit DK-TM4C123G

Kit DK-TM4C123G có các thông số cơ bản sau:

* Vi xử lý Tiva TM4C123GH6PGE.
* Cảm biến gia tốc 9- trục (accelerometer + gyro + compass).
* Hai cảm biến nhiệt độ
  + Internal microcontroller temperature sensor
  + External TMP20 temperature sensor
* Controller Area Network (CAN) transceiver.
* Bốn chân đầu vào Analog (0-20V).
* Microcontroller current shunt amplifier
* Màn hinh màu OLED 96 x 64.
* USB Micro-AB connector for Host/Device/OTG
* microSD card slot
* Năm nút bấm
* LED hỗ trợ cho người dung tại chân PG2
* Debug
  + In-Circuit Debug Interface (ICDI)
  + Standard 10-pin JTAG header (debug-out capable)
* Pin hỗ trợ chế độ Hibernate
* Nút Reset



*Hình 2.: sơ đồ khối cho Kit DK-TM4C123G*

2.3.3. Tổng quan về USB

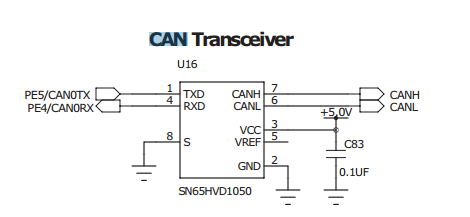
Kit DK-TM4C123G hỗ trợ giao tiếp USB Micro-AB

Sơ đồ chân:

|  |  |  |
| --- | --- | --- |
| GPIO Pin | Pin function | USB |
| PL6 | USB0DP | D+ |
| PL7 | USB0DM | D- |
| PB0 | USB0ID | ID |
| PB1 | USB0VBUS | USB VBUS |

2.3.4. Tổng quan về giao tiếp CAN trong Board DK-TM4C123G

Sử dụng chân PE4 và PE5 của TM4C123GH6PGE (module UART 5) để truyền nhận dữ liệu trong CAN BUS thông qua IC High-Speed CAN Transceceiver SN65HVD1050.



*Hình 2.: CAN Transceceiver*

Để sử dụng module CAN của Kit chúng kết nối hai chân CANH và CANL và BUS CAN cần kết nối.

**2.4. Kết luận chương**

Trong chương này chúng ta đã tìm hiểu được tương đối đầy đủ về giao thức CAN và USB (CDC) nhằm phục vụ cho mục đích mô phỏng node CAN.

**Tài liệu tham khảo**

1. Tài liệu giới thiệu về giao tiếp CAN của hãng Texas Instruments - Application Report SLOA101A–August 2002–Revised July 2008 (http://www.ti.com/lit/an/sloa101a/sloa101a.pdf).
2. Tài liểu mô tả về các chuẩn CAN của BOSCH - CAN Specification-1991, Robert Bosch GmbH, Postfach 50, D7000 Stuttgard 1. (<http://esd.cs.ucr.edu/webres/can20.pdf>).
3. USB Complete The Developer's Guide, Fifth Edition Jan Axelson
4. Datasheet của dòng chip Tiva™ TM4C123GH6PGE.
5. Tài liệu hướng dẫn sử dụng thư viện TivaWare™ Peripheral Driver Library
6. Tài liệu hướng dẫn sử dụng thư viện TivaWare™ USB Library