

## BÁO CÁO THỰC HÀNH BÀI 6

Môn học: **CHUYÊN ĐỀ THIẾT KẾ HỆ THỐNG NHÚNG 1**- Mã lớp: **CE437.N11**

Giảng viên hướng dẫn thực hành: **Phạm Minh Quân**

<b>Thông tin sinh viên</b>	<div>MSSV 20520211 20520219 20520597</div> <div>Họ và tên Trương Hữu Khang Nguyễn Linh Anh Khoa Phan Duy Thông</div>
<b>Link các tài liệu tham khảo</b> (nếu có)	<a href="https://piembsystech.com/security-access-service-identifier-0x27-uds-protocol/?utm_source=pocket_reader">https://piembsystech.com/security-access-service-identifier-0x27-uds-protocol/?utm_source=pocket_reader</a>
<b>Đánh giá của giảng viên:</b> + Nhận xét + Các lỗi trong chương trình + Gợi ý	

[Báo cáo chi tiết các thao tác, quy trình sinh viên đã thực hiện trong quá trình làm bài thực hành. Chụp lại hình ảnh màn hình hoặc hình ảnh kết quả chạy trên sản phẩm. Mô tả và giải thích chương trình tương ứng để cho ra kết quả như hình ảnh đã trình bày. Sinh viên xuất ra file .pdf và đặt tên theo cấu trúc: MSSV\_HoTen\_Labx\_Report.pdf (Trong đó: MSSV là mã số sinh viên, HoTen là họ và tên, x trong Labx là chỉ số của bài thực hành tương ứng)]

## MỤC LỤC

BÁO CÁO THỰC HÀNH BÀI 6.....	1
Câu 1. Viết chương trình để thực hiện Service \$27H – Security Access của UDS Protocol .....	3

## Câu 1. Viết chương trình để thực hiện Service \$27H – Security Access của UDS Protocol

Bài tập được thực hiện trên 2 board mạch NUCLEO-F303RE (Tester) và NUCLEO-F746ZG (ECU)

Phần chương trình của Tester:

```
while (1) {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    // create a 0x27 service request
    for (int i = 0; i < 8; i++) {
        Node_TxData[i] = 0x55;
    }
    Node_TxData[0] = 0x02;
    Node_TxData[1] = 0x27;
    Node_TxData[2] = 0x01; // SBF request seed

    HAL_CAN_AddTxMessage(&hcan, &Node_TxHeader, Node_TxData,
&Node_TxMailbox); // send request

    while (!new_message) { // waiting for response
        HAL_Delay(1);
        if (new_message) break;
    }
    if (message[0] == 0x27 + 0x40) { // check if positive response
        if (message[1] % 2 != 0) {
            seed = message[2];
            key = CalculateKey(seed);
            for (int i = 0; i < 8; i++) {
                Node_TxData[i] = 0x55;
            }
            Node_TxData[0] = 0x04;
            Node_TxData[1] = 0x27;
            Node_TxData[2] = 0x02; // SBF sendkey
            Node_TxData[3] = key;
            HAL_CAN_AddTxMessage(&hcan, &Node_TxHeader, Node_TxData,
&Node_TxMailbox);
            new_message = false;
        }
    }
    while (!new_message) { // wait for response
        HAL_Delay(1);
        if (new_message) break;
    }
    if (message[0] == 0x27 + 0x40) {
        if (message[1] % 2 == 0) {
            printf("Access granted\n");
            HAL_GPIO_TogglePin(LD2_GPIO_Port, LD2_Pin);
            new_message = false;
        }
    }
    HAL_Delay(5000);
}
```

Phần chương trình của ECU:

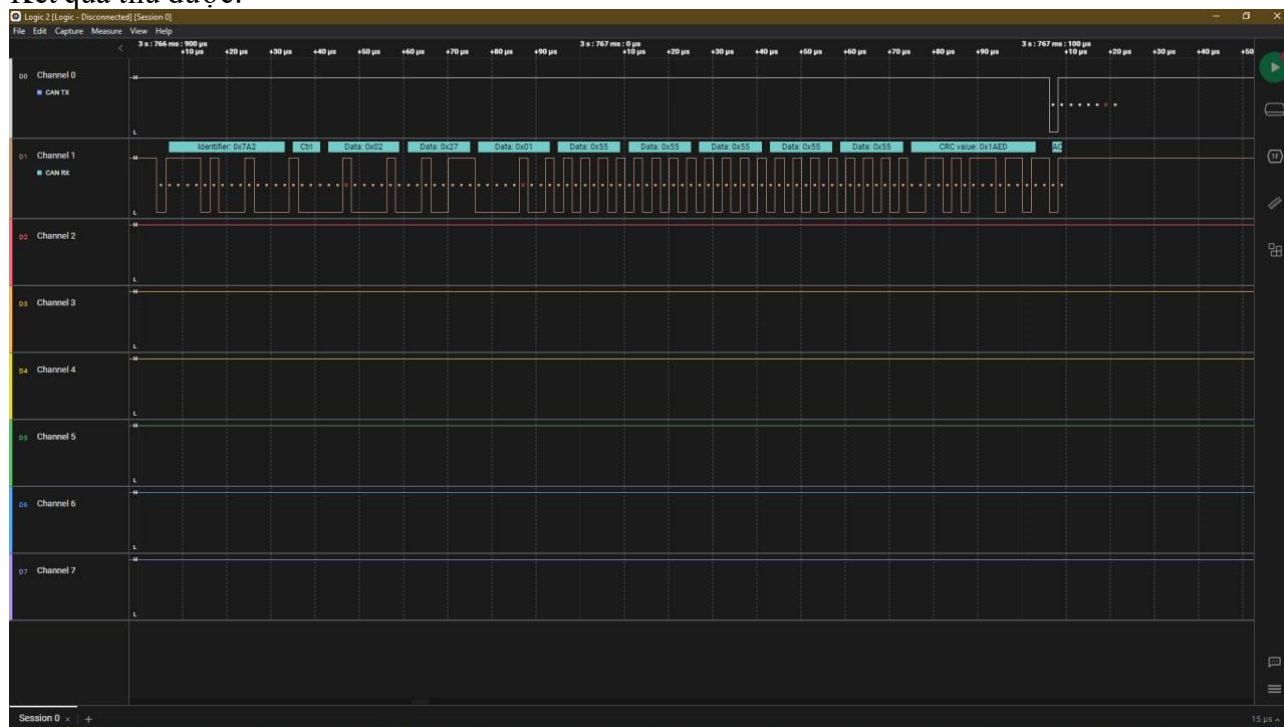
```
while (1) {
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    while (!new_message) {
        HAL_Delay(1);
        if (new_message) {
            break;
        }
    }
    if (message[0] == 0x27) {
        if (message[1] % 2 == 1) { // check if SBF Seed or Key
            seed = GenerateRandomNumber(0, 255); // generate random seed
            PrepareMessage();
            Node_TxData[0] = 0x03;
            Node_TxData[1] = 0x27 + 0x40;
            Node_TxData[2] = 0x01;
            Node_TxData[3] = seed;
            new_message = false;
            HAL_CAN_AddTxMessage(&hcan1, &Node_TxHeader, Node_TxData,
&Node_TxMailbox); // send seed
            key = CalculateKey(seed);
            printf("Seed: %d, Key: %d\r", seed, key);
            new_message = false;
        } else if (message[1] % 2 == 0) { // check if SBF Seed or Key
            if (message[2] == key) { // check key
                PrepareMessage();
                Node_TxData[0] = 0x02;
                Node_TxData[1] = 0x27 + 0x40;
                Node_TxData[2] = 0x02;
                unlocked = true;
                __HAL_TIM_SET_COUNTER(&htim2, 0); // start counting
                printf("Unlocked\r");
                HAL_GPIO_TogglePin(LD1_GPIO_Port, LD1_Pin);
                HAL_CAN_AddTxMessage(&hcan1, &Node_TxHeader, Node_TxData,
&Node_TxMailbox);
                new_message = false;
            }
        }
    }
    if (unlocked) {
        // accept any request
    } else {
        // reject any request if not unlocked
    }
}
```

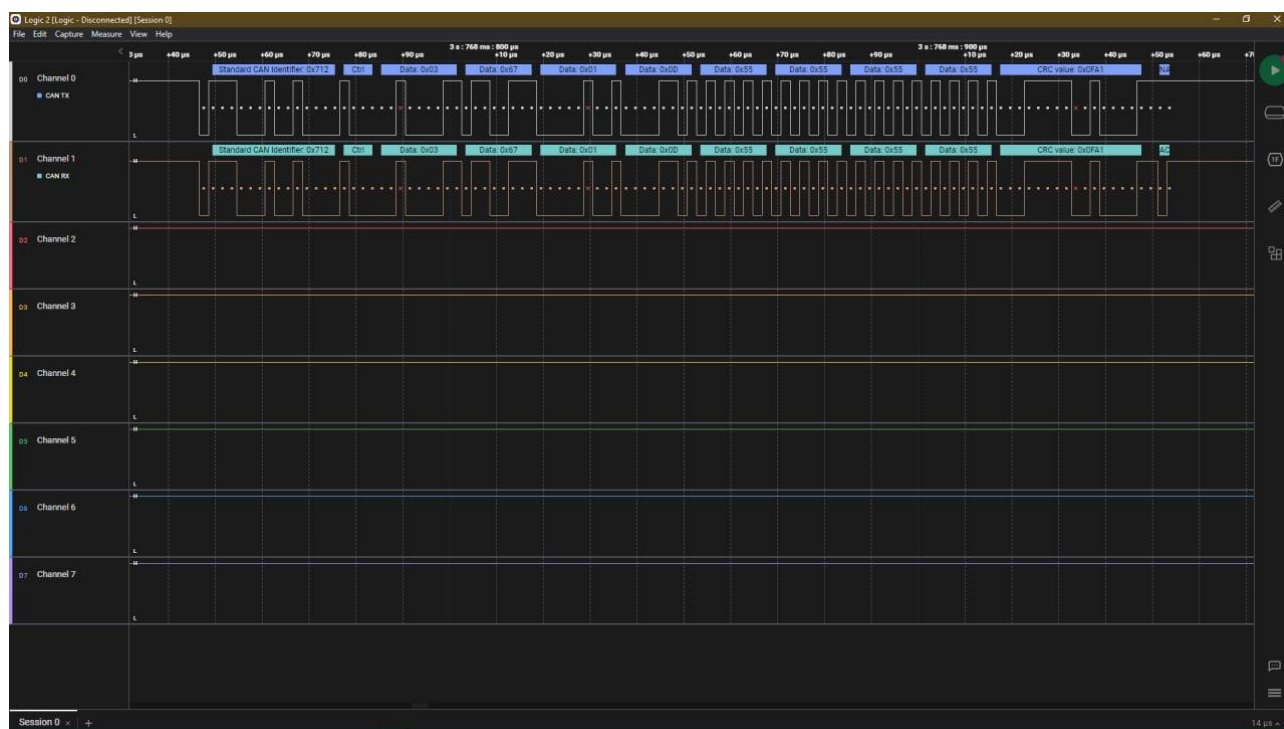
Giải thích:

Tester sẽ gửi Request \$27H đến ECU, sau đó ECU sẽ tạo sinh số ngẫu nhiên và gửi trả về cho Tester đồng thời tính toán ra “key”, sau khi Tester nhận được số seed sẽ tính toán ra key và gửi cho ECU. Khi nhận được đúng “key” thì ECU sẽ gửi gói tin xác nhận về cho Tester. Sau khi đã unlock ECU sẽ thực hiện các yêu cầu của Tester, sau một khoảng thời gian sau khi đã unlock, nếu không nhận được bất cứ request nào từ Tester, ECU sẽ tự động lock lại sau một khoảng thời gian thông qua một Timer Interrupt.

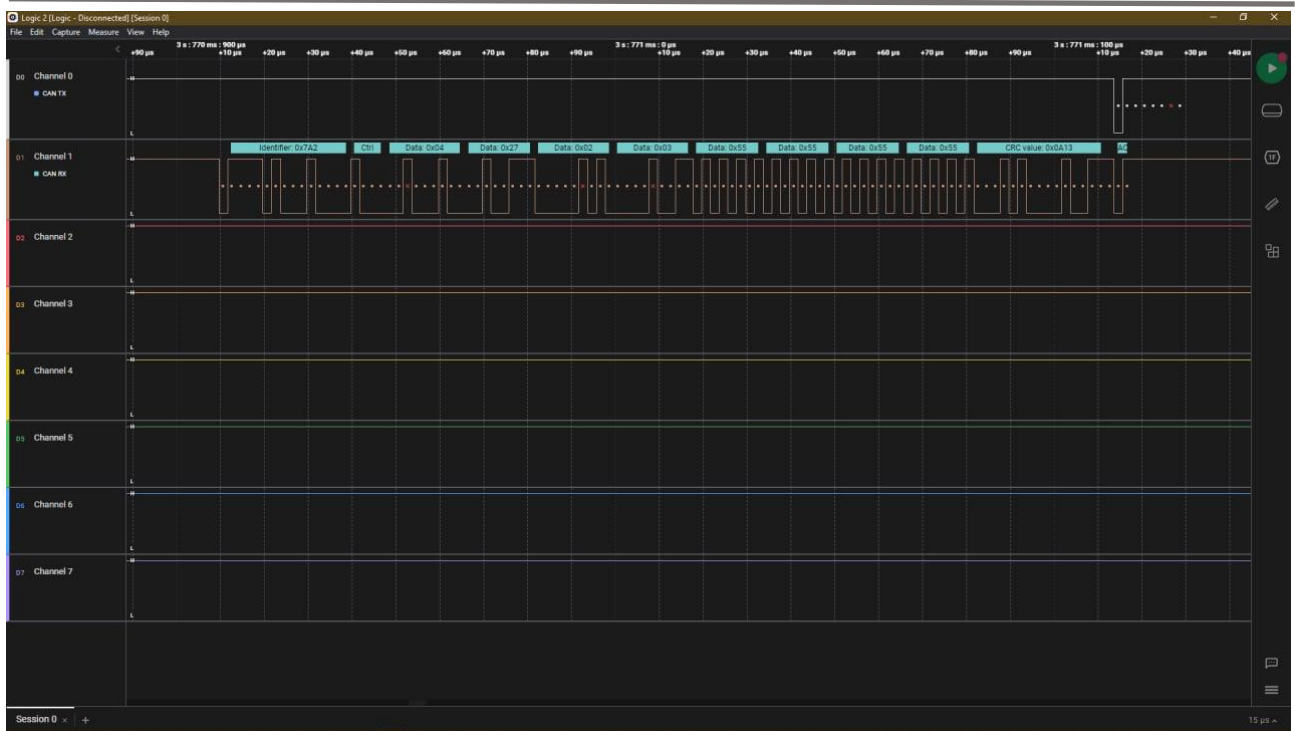
Kết quả thu được:



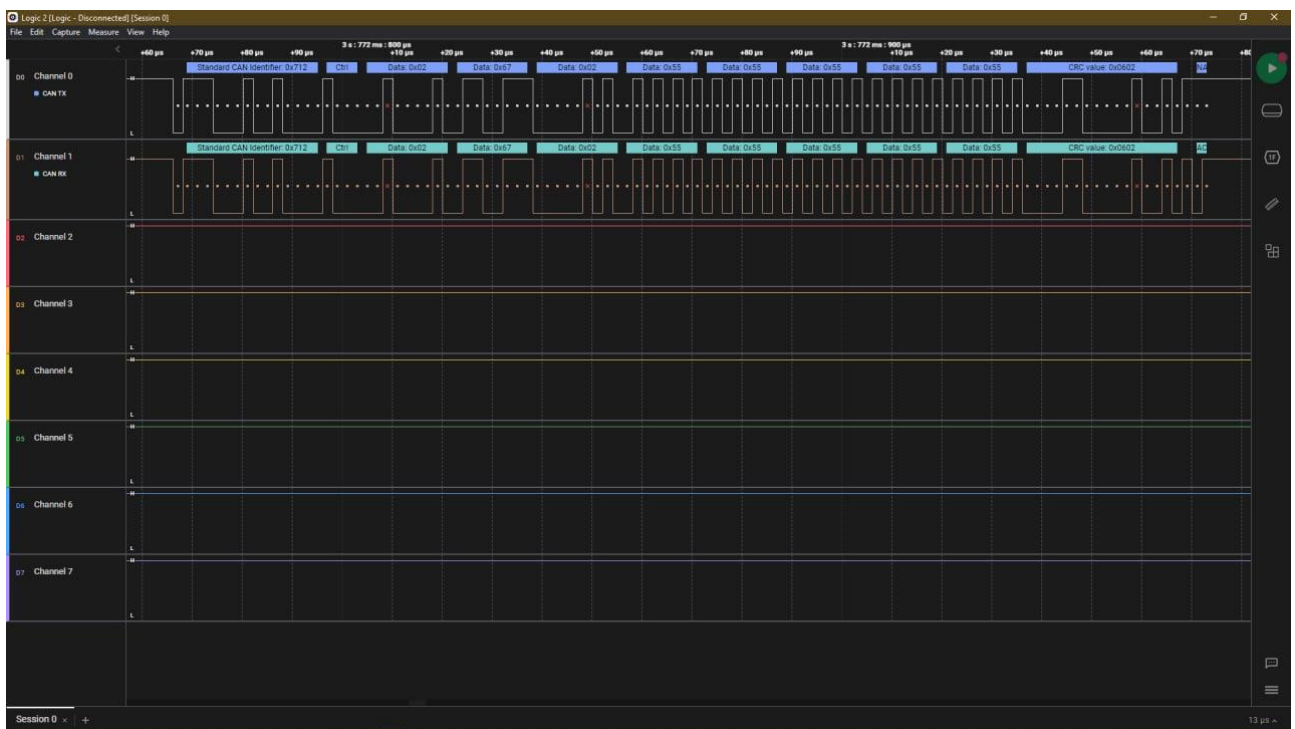
Hình 1. Gói tin Request Seed của Service \$27H



Hình 2. Gói tin Reponse kèm với Seed từ ECU



Hình 3. Gói tin chứa Key đã tính toán từ Tester



Hình 4. Gói tin xác nhận Key đã đúng từ ECU