

NODEMCU QURILMASIDA DASTURLASH

Eraliyev Nosirbek Fazliddin o'g'li

Andijon mashinasozlik instituti

1-kurs magistrant, mexatronika va robototexnika yo'nalishi

Annotatsiya. Ushbu maqolada nodemcu qurilmasida dasturlash haqida so'z yuritilgan. NodeMCU modullari bizning Lua 5.1 va Lua 5.3 ish vaqtlariga nisbatan kompilyatsiya qilish uchun mo'ljallangan. NRM bu erda oqibatlar va cheklovlarni muhokama qiladi.

Kalit so'zlar: nodemcu qurilmasi, dasturlash, modul, ishlab chiqaruvchi.

Standart Lua ish vaqti Lua 5.3 Yo'naltiruvchi qo'llanmada ("LRM") va xususan LRM 6.3 bo'limida tavsiflanganidek, bir nechta Lua funksiyalari va xususiyatlarini inkapsulatsiyalash jadvalida belgilashi mumkin bo'lgan ikkala Lua modulini qo'llab-quvvatlaydi. Lua, shuningdek, ishlab chiquvchilarga kompilyatsiya qilingan C-da modullarni amalga oshirishga ruxsat berish uchun C API-ni taqdim etadi. NodeMCU ishlab chiquvchilari, shuningdek, ushbu standart API yordamida o'zlarining C modullarini ishlab chiqishlari va o'zlarining mikrodasturlariga qo'shishlari mumkin, NodeMCU ilovasi NodeMCU Reference Manual ("NRM") da hujjatlashtirilganidek, standart Lua-ni kengaytiradi. Mavjud C modullarini ishlab chiqmoqchi yoki o'zgartirmoqchi bo'lgan ishlab chiquvchilar LRM, PiL va NRM ga kirishlari va ushbu havolalar bilan tanishishlari kerak. Bular asosiy havolalar; va ushbu hujjat ushbu tarkibni takrorlamaydi, balki uni to'ldirish uchun NodeMCU-ga xos ma'lumotlarni taqdim etadi. C modullarini ishlab chiqish nuqtai nazaridan, standart Lua-da modullarni ishlab chiqishdan juda kam farq bor. Barcha standart Lua kutubxona modullari (bit, koroutin, disk raskadrovka, matematika, string, jadval, utf8) Lua uchun C API-dan foydalanadi va NodeMCU versiyalari NRM kengaytmalaridan foydalanish uchun yangilangan. shuning uchun ularning manba kodi ko'rish va uslub shablonlari sifatida foydalanish uchun mavjud.

Asosiy funktsional o'zgarish shundaki, NodeMCU statik const deklaratsiyasidan foydalangan holda modul manbasida statik ravishda e'lon qilinishi mumkin bo'lgan ROTable deb nomlanuvchi Jadval turidagi faqat o'qish uchun mo'ljallangan kichik sinfni qo'llab-quvvatlaydi. Shuningdek, ROTable kalitlari uchun amal turlari va qiymatlari statik deklaratsiyaga mos kelishini ta'minlash uchun cheklovlar mavjud; va shuning uchun ROTables kod maydonida (va shuning uchun IoT qurilmasidagi flesh xotirada) saqlanadi. Shunday qilib, standart Lua jadvallaridan farqli o'laroq, ROTables RAM resurslarini egallamaydi.

Bundan tashqari, standart Lua-dan farqli o'laroq, C modullarini ro'yxatdan o'tkazish uchun ikkita global ROTables ishlatiladi. Shunga qaramay, statik deklaratsiya makroslari va bog'lovchi "sehrli" (makrolik filtrlardan foydalanish va bog'lovchi bo'lim ko'rsatmalari) yaratish jarayonida ushbu ROTable-larni birlashtirishga olib keladi va ularning barchasi ROTable-ga asoslanganligi sababli, modullarning proshivka tuzilmalariga integratsiyalashuvi va ularning Lua ilovalarini bajarishdan kirish RAMga asoslangan ma'lumotlar tuzilmalariga emas, balki kod maydoniga bog'liq. Esda tutingki, C modullarini dinamik yuklash ESP SDK da qo'llab-quvvatlanmaydi, shuning uchun har qanday kutubxonani ro'yxatdan o'tkazish proshivka yaratishda foydalanilgan manbaga kompilyatsiya qilinishi kerak. Bizning yondashuvimiz oddiy, moslashuvchan va standart Lua yondashuvining RAM qo'shimcha xarajatlaridan qochadi. Maxsus ROTable ROM bu yondashuvning asosidir. Global muhit jadvali ushbu ROM jadvaliga havola qiluvchi __index metametodiga ega. Shunday qilib, global jadvalga nisbatan xom bo'lmagan har qanday qidiruvlar ROMga qarshi ham hal qilinadi. Barcha asosiy Lua funktsiyalari (masalan, chop etish) va har qanday C kutubxonalari (NodeMCU standartlariga muvofiq yozilgan) ROM jadvalida yozuvga ega va shuning uchun global ko'rinishga ega. Ushbu yondashuv ishlab chiquvchilarga standart Lua mexanizmlaridan foydalanishga to'sqinlik qilmaydi, aksincha u oddiy past RAM foydalanish muqobilini taklif qiladi. NODEMCU_MODULE makrosi har bir modulda uni ROT ROMdagi yozuvda ro'yxatdan o'tkazish uchun ishlatiladi.

Shuningdek, u ikkinchi (yashirin) ROMentry ROTable-ga yozuvni qo'shadi. - Barcha ROM yozuvlari global miqyosda hal qilinadi - Lua ish vaqti ishga tushirish vaqtida ROMentry ROTable-ni skanerlaydi va u ushbu jadvaldagi NULL bo'lmagan CFunction qiymatlarini bajaradi. Bular, agar kerak bo'lsa, C modullariga bir

martalik ishga tushirish funksiyalariga ulanish imkonini beradi. E'tibor bering, agar mos keladigan `LUA_USE_MODULES_modname` makrosi aniqlangan bo'lsa, standart ishlab chiqarish proshivka tuzilmasi ichidagi ilovalar/modullar jildida topilgan har qanday modullarni o'z ichiga oladi. Ushbu ta'riflar an'anaviy ravishda umumiy foydalanuvchi `modules.h` faylida o'rnatiladi va bu amaliyot NodeMCU tarqatilishiga qo'shilgan har qanday foydalanuvchi tomonidan yuborilgan modullar uchun majburiydir. Biroq, bu ishlab chiquvchilarga o'zlarining mahalliy modullarini ilova / modullar jildiga qo'shishga va oddiygina mos keladigan `LUA_USE_MODULES_modname` inline ni belgilashga to'sqinlik qilmaydi. Ushbu so'l + bog'lovchi yondashuvi `luaL_reg` deklaratsiyasi va `luaL_openlib()` dan foydalanish zaruratini yo'qotadi va bunga loyiha tomonidan qabul qilingan ilova/modul fayllarida ruxsat berilmaydi.

Shunday qilib, NodeMCU C kutubxona moduli odatda standart Lua kutubxona modullari bilan parallel bo'lgan standart tartibga ega va Lua ish vaqtiga kirish uchun bir xil C API dan foydalanadi. Tashqi resurslarga kirishni hal qilish uchun `#include` bloki. Barcha modullar `"module.h"` va `"lauxlib.h"` uchun yozuvlarni o'z ichiga oladi. Ular Lua manba katalogidagi boshqa `LXXX.h` tarkibiga havola qilmasliklari kerak, chunki ular Lua ish vaqti uchun maxfiydir. Ulardan keyin C standart ish vaqti, tashqi ilovalar kutubxonalari va har qanday SDK resurs sarlavhalari kerak bo'lishi mumkin. Esda tutingki, biz `<stdlib.h>` va hokazolar uchun C standart ish vaqti API'sidan foydalanishni tavsiya qilsak ham, SDK faqat ushbu API'ning noto'g'ri hujjatlashtirilgan quyi to'plamini qo'llaydi, shuning uchun siz bog'lovchi xatoliklarga duch kelishingiz va bu holatda qayta kodlashingiz kerakligini yodda tuting. C modulining yagona tashqi interfeysi Lua ish vaqti va uning `NODEMCU_MODULE` ilgaklari orqali bo'lishi kerak. Shuning uchun barcha funktsiyalar va resurslar statik deb e'lon qilinishi va modul uchun shaxsiy bo'lishi kerak. Ularni ishlab chiquvchining xohishiga ko'ra buyurtma qilish mumkin, albatta, C miqyosi qoidalariga rioya qilish uchun tegishli oldingi deklaratsiyalarga ehtiyoj bor. Modul usullari odatda Lua standart statik `int somefunc(lua_State *L) { ... }` shablonidan foydalanadi. ROT jadvallari odatda modul oxirida e'lon qilinadi va NRMda tasvirlangan LROT makroslaridan foydalanish va oldinga yo'naltirish zaruratini kamaytirish uchun. Shuni yodda tutingki, ROTables faqat string kalitlari va qiymat turlarining statik deklaratsiyasini qo'llab-quvvatlaydi: C funktsiyasi, engil foydalanuvchi ma'lumotlari, Raqamli, ROTable. ROTable-da ROTable meta-jadvallari ham bo'lishi mumkin. ROTable qidiruv algoritmi ROTable yozuvlarining oddiy chiziqli skanerlashi bo'lsa-da, ish vaqti ROTable kirishlarining LRU keshini ham saqlaydi, shuning uchun odatda ROTable kirishlarining 95% dan ortig'i chiziqli skanerlashni chetlab o'tadi va tegishli yozuvga to'g'ridan-to'g'ri kirishni amalga oshiradi. ROTables ham o'rtacha engil va Lua ish vaqtiga yaxshi integratsiyalashgan, shuning uchun oddiy metametodni qayta ishlash yaxshi ishlaydi. Bu shuni anglatadiki, ishlab chiquvchilar indeks funktsiyasi orqali boshqa kalit va qiymat kiritilgan yozuvlarni amalga oshirish uchun `__index` usulidan foydalanishlari mumkin.

NodeMCU modullari bizning Lua 5.1 va Lua 5.3 ish vaqtlariga nisbatan kompilyatsiya qilish uchun mo'ljallangan. NRM bu erda oqibatlar va cheklovlarni muhokama qiladi. Ammo shuni yodda tuting: Biz ko'plab yangi Lua 5.3 funksiyalarini NodeMCU Lua 5.1 API-ga o'rnatilgan, shuning uchun siz modullaringizni kodlash uchun 5.3 API-dan foydalanishingiz mumkin. NRM yana istisnolarni qayd etadi, bunda sizga variant kodi kerak bo'ladi yoki o'zingizni 5.3 ish vaqti bilan cheklashga qaror qilasiz. Bu oxirgi holatda eng oddiy yondashuv `#if LUA_VERSION_NUM != 503` 5.3 tarkibini o'chirish uchun 5.1 tuzilmasi kompilyatsiya qilish va bog'lash uchun. E'tibor bering, hozirda ilova / modullar jildidagi barcha modullar Lua 5.1 va 5.3 muhitlariga qarshi kompilyatsiya qilinadi va ishlaydi. Lua 5.3 butun son (32 bitli imzolangan tamsayı sifatida saqlanadi) va float (32 bitli bitta aniqlikdagi float sifatida saqlanadi) uchun alohida subtiplari bilan barcha raqamlar uchun 32 bitli tasvirdan foydalanadi. Bu Lua 5.1 butun soni bilan bir xil RAM saqlash zichligiga erishadi, agar qulay bo'lsa, suzuvchi nuqtadan foydalanishni yo'qotmaydi. Shuning uchun biz alohida Integer 5.3 qurish variantiga ega bo'lishning foydasi yo'q deb qaror qilingan. Ishlab chiquvchilarga kodning aniqligini kamaytirish uchun `luaL_API` chaqiruvlarining to'liq to'plamidan foydalanishni tavsiya qilamiz. Biz, shuningdek, qayta qo'ng'iroqlarni qo'llab-quvvatlash uchun Lua registridan foydalanganda oddiy va ishlash vaqtini yaxshilaydigan bir nechta registrga kirishni optimallashtirishni qo'shilgan. `luaL_reref()` mavjud ro'yxatga olish kitobi ma'lumotnomasini joyida almashtiradi (yoki kerak bo'lganda yangisini yaratadi). `luaL_unref()` va `luaL_ref()` konstruktsiyasidan kamroq kod va tezroq bajarilishi. `luaL_unref2()` unrefni bajaradi va statik `int` kancasini `LUA_NOREF` ga o'rnatadi.

Adabiyotlar

1. Мойдинова, Э. К. (2021). РАЗВИТИЕ ПРОФЕССИОНАЛЬНОЙ КОМПЕТЕНЦИИ БУДУЩИХ УЧИТЕЛЕЙ АНГЛИЙСКОГО ЯЗЫКА. In СИСТЕМА МЕНЕДЖМЕНТА КАЧЕСТВА В ВУЗЕ: ЗДОРОВЬЕ, ОБРАЗОВАННОСТЬ, КОНКУРЕНТОСПОСОБНОСТЬ (pp. 213-217).
2. Мойдинова, Э. К. (2019). Дистанционное обучение в современном вузе и использование технологии "e-learning". NovaInfo. Ru, (104), 70-72.
3. Мойдинова, Э. К. (2018). Методы преподавания фразеологических единиц для обогащения словарного запаса. NovaInfo. Ru, 1(94), 137-140.
4. Мойдинова, Э. К. (2018). ИСПОЛЬЗОВАНИЕ ИКТ В РАЗВИТИИ ПОЗНАВАТЕЛЬНОГО ИНТЕРЕСА ПРИ ИЗУЧЕНИИ АНГЛИЙСКОГО ЯЗЫКА В НАЧАЛЬНОЙ ШКОЛЕ. ББК 81.411. 2 Р89, 225.
5. УЗБЕКИСТАН, К. Р. РОЛЬ МУЛЬТИМЕДИЙНЫХ ТЕХНОЛОГИЙ В ОБУЧЕНИИ ИНОСТРАННЫМ ЯЗЫКАМ Материалы республиканской научно-практической конференции.
6. Kamariddinovna, M. E. THE ROLE OF INTERCULTURAL COMMUNICATION IN THE TRAINING FOR FUTURE SPECIALIST OF DIFFERENT FIELDS. Zbiór artykułów naukowych recenzowanych., 2, 169.