Arduino

Какво е Arduino

Arduino е електронна платформа с отворен код, базирана на лесен за използване хардуер и софтуер. Arduino дъските могат да четат входове - светлина върху сензор, пръст върху бутон или съобщение в Twitter - и го превръщат в изход - активиране на двигател, включване на светодиод, публикуване на нещо онлайн. Можете да кажете на вашия съвет какво да прави, като изпратите набор от инструкции до микроконтролера на дъската. За целта използвате езика за програмиране на Arduino (базиран на Wiring) и Arduino Software (IDE), базиран на Processing.

През годините Arduino е мозъкът на хиляди проекти, от ежедневни обекти до сложни научни инструменти. Световна общност от производители - студенти, хобисти, художници, програмисти и професионалисти - се е събрала около тази платформа с отворен код, техните приноси са добавили към невероятно количество достъпни знания, които могат да бъдат от голяма полза както за начинаещи, така и за експерти.

Ардуино е роден в Ivrea Interaction Design Institute

като лесен инструмент за бързо създаване на прототипи, насочен към студенти без опит в електрониката и програмирането. Веднага след като достигна до по-широка общност, бордът на Arduino започна да се променя, за да се адаптира към новите нужди и предизвикателства, диференцирайки своето предложение от прости 8-битови дъски до продукти за IoT приложения, носими, 3D печат и вградени среди. Всички дъски на Arduino са напълно отворен код, даващи възможност на потребителите да ги изграждат независимо и в крайна сметка да ги адаптират към техните специфични нужди. Софтуерът също е с отворен код и се разраства чрез приноса на потребителите по целия свят.

Защо Ардуино?

Благодарение на своето просто и достъпно потребителско изживяване, Arduino е използван в хиляди различни проекти и приложения. Софтуерът Arduino е лесен за използване за начинаещи, но достатъчно гъвкав за напреднали потребители. Работи се на Mac, Windows и Linux. Учителите и студентите го използват за изграждане на нискотарифни научни инструменти, за доказване на принципите на химията и физиката или за започване на програмиране и роботика. Дизайнерите и архитектите изграждат интерактивни прототипи, музикантите и художниците го използват за инсталации и да експериментират с нови музикални инструменти. Създателите, разбира се, го използват за изграждането на много от проектите, изложени на Faire Maker. Arduino е ключов инструмент за научаване на нови неща. Всеки - деца, любители, артисти, програмисти - може да започне да се занимава само следвайки стъпка по стъпка инструкциите на комплект или да споделя идеи онлайн с други членове на общността на Arduino.

Има много други микроконтролери и микроконтролерни платформи, достъпни за физически изчисления. Parallax Basic Stamp, BX-24 на Netmedia, Phidgets, MIT's Handyboard и много други предлагат подобна функционалност. Всички тези инструменти вземат разхвърляните подробности за програмирането на микроконтролери и ги увиват в лесен за използване пакет. Arduino също опростява процеса на работа с микроконтролери, но предлага известно предимство за учители, студенти и заинтересовани любители пред други системи:

Евтин - Arduino платките са сравнително евтини в сравнение с други платформи за микроконтролери. Най-малко скъпата версия на модула Arduino може да бъде сглобена на ръка и дори предварително сглобените Arduino модули струват по-малко от $ 50

Крос-платформа - Софтуерът Arduino (IDE) работи на операционни системи Windows, Macintosh OSX и Linux. Повечето микроконтролери са ограничени до Windows.

Проста, ясна програмна среда - Софтуерът Arduino (IDE) е лесен за използване за начинаещи, но достатъчно гъвкав и за напреднали потребители, за да се възползва от тях. За преподавателите удобно се основава на програмната среда за обработка, така че студентите, които се учат да програмират в тази среда, ще бъдат запознати с начина, по който работи Arduino IDE.

С отворен код и разширим софтуер - Софтуерът Arduino се публикува като инструменти с отворен код, достъпни за разширение от опитни програмисти. Езикът може да бъде разширен чрез C++ библиотеки и хората, които искат да разберат техническите подробности, могат да направят скока от Arduino към езика за програмиране на AVR C, на който е базиран. По същия начин можете да добавите AVR-C код директно във вашите програми на Arduino, ако искате.

Отворен код и разтегателен хардуер - Плановете на платките на Arduino са публикувани под лиценз Creative Commons, така че опитни дизайнери могат да направят своя собствена версия на модула, като го разширят и подобрят. Дори сравнително неопитни потребители могат да създадат версията на модула, за да разберат как работи и да спестят пари.

FireBase realtime database

React Native

React Native е рамка за мобилни приложения с отворен код, създадена от Facebook. Използва се за разработване на приложения за Android, iOS, Web и UWP, като дава възможност на разработчиците да използват React заедно с възможностите на родната платформа.

React Native позволява на разработчиците да създават кросплатформени приложения, които изглеждат и се чувстват изцяло Native, тъй като използва JavaScript компоненти, които са изградени както на iOS, така и на Android компоненти. Това спестява много време и пари, тъй като разработчиците не трябва да създават едно и също приложение за множество платформи. Освен това е по-лесно да се поддържа приложение, създадено с React Native, тъй като има само една кодова база. React Native използва основни блокове за Android и iOS, за да компилира Native приложения и за двете платформи в JavaScript. Това прави работата с кодовата база по-лесна, а добавянето на нови функции също е опростено, дори докато приложението работи.

A picture containing monitor

Description automatically generated

Нека да разгледаме работния процес на React Native, както следва:

Първоначално UI нишката започва да зарежда JS пакети и изпълнения в приложението. Тези две нишки никога не комуникират директно и блокират взаимно пътя си в резултат на синхронизация.

UI нишката и JS нишката никога не споделят и не работят върху едни и същи данни. Те предпочитат да обменят сериализирани съобщения.

Докато започва визуализацията, Reconciler започва "диференциация", когато генерира нов V-DOM и модифицираното оформление се изпраща към shadow нишка, която генерира shadow nodes.

След изчисляване на оформлението, той изпраща параметъра към нишката на потребителския интерфейс, защото единственият основен поток може да изобрази нещо на екрана.

Потребителският интерфейс ще се представи, когато нишката на сенките ще изпрати генерирано оформление към нишката на потребителския интерфейс.

Има 4 вида нишки в реагиращите местни приложения:

В реагиращото местно приложение нишките играят основна роля, като позволяват отделни задачи да се изпълняват заедно, докато една нишка изпълнява една задача, друга нишка може да изпълнява някаква друга отделна задача по паралелен начин.

Основна нишка:

Известена е също като UI нишка, която се използва за естествено изобразяване на iOS или Android UI. Той се създава автоматично в началото на приложенията, тъй като основната нишка ще породи всички долни нишки.

UI нишката е и последната нишка, която завършва изпълнението, като изпълнява различни задачи за затваряне.

JavaScript нишка:

Той ще стартира основния логически код на JavaScript, включително сензорни събития, API обаждания и много други. Всички актуализации на собствените изгледи се пакетират и в края на всеки цикъл на събитието в нишката на JavaScript ще бъдат изпратени директно към основната страна.

За по-добра ефективност JavaScript нишката изпраща пакетните актуализации в потребителския интерфейс, преди следващия кадър да представи крайния срок.

Има и едно изключение от JS нишката, че родният изглед, който работи в нишката на потребителския интерфейс, не е блокиран поради бавна js нишка. напр .: ScrollView работи в UI нишка.

Native Modules нишка:

Native модулите обикновено се разпространяват като npm пакети, с изключение на типичните JavaScript файлове и се използват за достъп до специфичния за платформата API.

Render нишка:

Това е специална нишка, която изпраща команди към графичния процесор и използва за генериране на действителни OpenGL команди за изготвяне на потребителски интерфейс.

HTML

CSS

Javascript

React